IT-214 DBMS

DBMS Final Project Documentation
Labor Management System

Group 6, Section 10, Team Id 10

Team members -

Vansh Ashish Parikh - 202001011
Dalwala Fenil Rajeshkumar – 202001130

Index

IV Conversion of Final ER-Diagram to Relational Model

1. Relational Schema
2. DDL scripts

V Normalization and Schema Refinement.

VI SQL: Final DDL Scripts, Insert statements, 20 SQL Queries with Snapshots of the output of each query.

1. Modified DDL Scripts
2. Simple Queries
3. Complex Queries
4. Function
5. Trigger

V Website Designing

1. Home page
2. Database Details
3. Sorting on table
4. Update operation
5. Delete Operation
6. Insert Operation
7. Running Custom Queries

City Labor Management System

1. Description/Case Study of the Problem Domain

1.1 Purpose

The purpose of this document is to provide a prototype of a management system which assists the district labor supervisors and their management team to efficiently manage the wages of the laborers working under each of them, as well as helps the city administration to establish a centralized control over district labor management. Its requirements are with respect to maintaining the records of laborers working in different districts under various supervisors and the type of wage he is being paid (weekly/daily), record of daily work being done by a laborer.

1.2 Intended Audience and Reading Suggestions

This is the wage management system of laborers and it is restricted within the city itself the main beneficiary of this system is none other then the supervisor of laborers and work management team.This system will influence every laborer of city because they are main subject of this document still they are the audience of this system as a laborer can verify its salary and it will ensure him that it salary will not be governed by an individual entity instead it is deployed by particular structure.

In addition it will also help any industry in which laborers are working because the structure also tracks the data of which works are done by laborers.

1.3 Product Scope

This Database system comes with its intent to ease the wage distribution process among workers and supervisors of the city which obviate any types of conflict. Without any doubt we can say that this product is much more scalable and it can influence the whole city labor management in a positive direction. It binds every single person involved in the city labor system into a single database and helps in efficiently managing the wages of laborers of the city. It can also reduce the complaints of the laborers not getting their wages regularly as decided as the system will be updated systematically such that the integrity remains intact.

1.4 Description

The main objective of the City Labor Management System is to maintain daily wages, labor and supervisor details. From daily anonymous attendance to regular employee attendance, inspecting the safety and standards to work on the field for workers to avoid unwanted damages. Track work orders for various sites and the various locations at a single view.

According to the problem, the required wage management system should have a focus on the following main parameters or we can alternately say these are main organizational issues we are keeping in mind while developing the system/product :

- **Proper and correct payment of wages:** There should be a concept of minimum wage in our system which will be paid to laborers in any case, furthermore it will be reviewed annually by district administrators. Wages need to be paid on

time without any unavoidable circumstances. Workers should have direct access to their wages and it will be given to their bank account. Workers are able to freely access their original identity card, work permits and travel documents at any time.All the working hours are paid, including overtime. Payment of overtime-hours for different periods (regular days, rest days and holidays) with legal required rates. overtime-hours should be within the legal limit.Workers are provided one day off in seven days. No discrimination - gender, sex, religion etc. No child labour .No forced labour.Induction training for workers and supervisors.

- **A progressive and coherent pay system:** Such a payment system is necessary for the growth/progress of any organization in which the workers are paid purely based on their work hours & skills required in the work they are doing. In addition to this, the legal benefits should also be given to the workers like insurance & factory monetary benefits. Payments of wages should be decided by keeping in mind the inflation & cost of living as well as other factors like age of laborer & his economic condition. Other than this, overtime wages should be paid if required. Incentive System should be fair enough to encourage workers.

- **Proper Mechanism of Workers Involvement:** There should be a channel for workers involvement on implementation of wage policies and for increasing wage & social awareness of workers.

- Create a personalized database for the laborers and their categories on the basis of work type, wage type & districts.
- Track attendance for both the daily wage workers and weekly wage workers.
- Allot work orders to laborers on daily and monthly basis through a centralized cloud system.
- Inspect the laborers on work sites through IoT attendance system.

- **Stay aware of the laborer's performance on-site with their shifts, overtime, in time, out time & total working hours.**

The entities and relations that we are planning to implement include:

1) **Laborer:** contains the information about all the workers/laborers of the city like his ID, name, age, address, district & his type of wage type (hourly/weekly).

2) **Supervisor:** contains the information about all the supervisors and it has the following attribute fields like their supervisor's id, their name,their addresses information about their district.

3) **District :** It contains the mapping of dist_id to the names of districts.

4) **Hourly & Daily Wage Rate :** It specifies the hourly wage rates as well as daily wage rates for a particular work type.

5) **Work and Wage Table :** After this we need to show which workers are assigned to which Supervisor, this table will help us to relate between supervisor and worker and It contains general information about supervisor id, laborer id, wage type which means hourly or weekly for laborers.

6) **Hourly Earning :** It is specifically designed for laborers earning hourly wages and it will be updated on a daily basis, it contains attributes like laborer's id,type of wage, how much hour they work per day, their daily earning which is derived field and multiplication of hours they worked and rage of wage per work.

7) **Weekly Earning:** This table is specifically designed for laborers earning weekly wages and it will be updated on a weekly basis. It includes characteristics like (Monday, Tuesday, Wednesday, …) which represent presence/absence (boolean domain) of the laborer on that particular day and total earnings are calculated on the basis of number of days present in a week multiplied by wage rate of that work type.

8) **General Visitors:** The general visitors of the system will have to first submit their basic information to access the data in the database and they will be able to access only a part of the database which will be given access according to the policies & procedures.

9) **Database System Managers:** This includes the technical team which will manage the backend & frontend of the system as well will be responsible for taking the time-to-time backup and maintenance of the database. Their information will also be stored as they are also the part of a large system involved in this.

Each user of the system (other than general visitors) will have a login id (laborer id, supervisor id, administrator id) & password which ensures the security of data as the users will be accessing the integral part of the database.

1.5 Schema

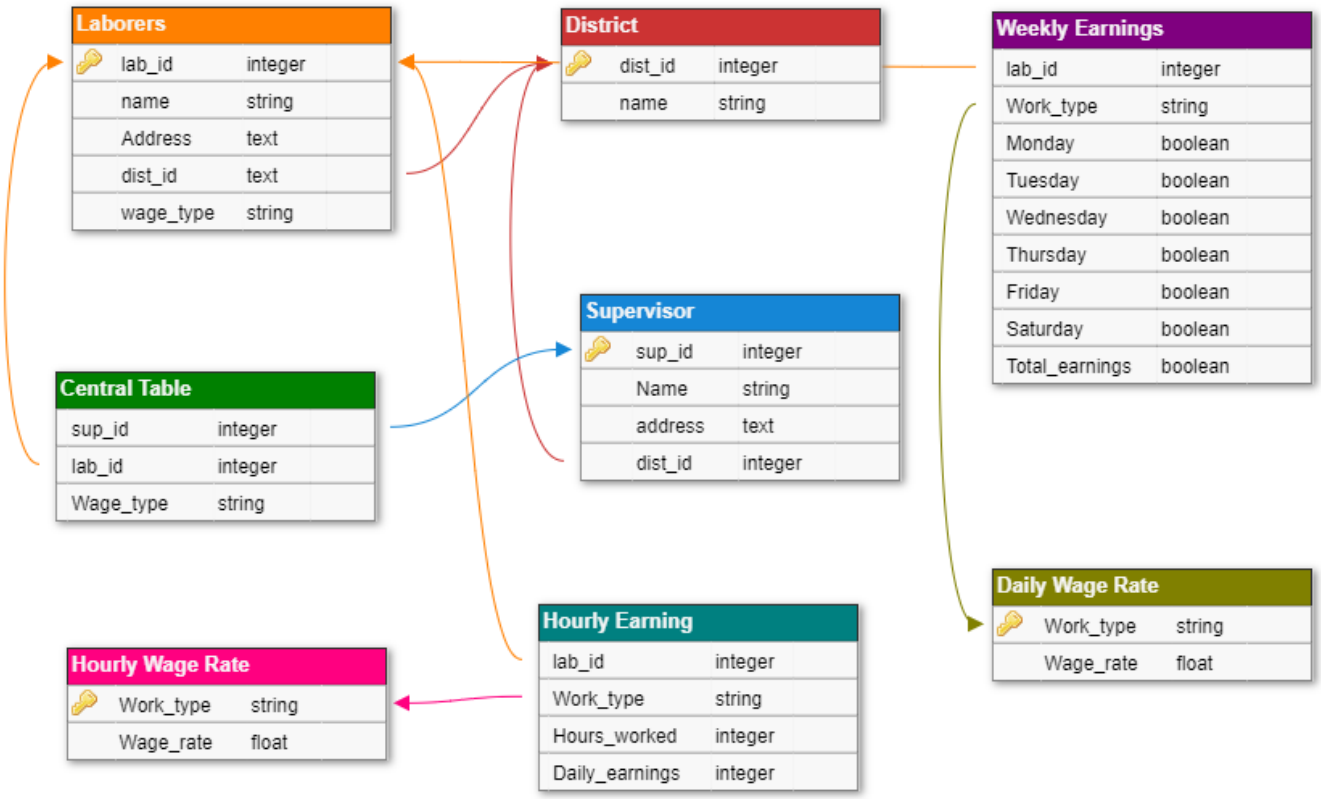


Table Requirements:

The tables that we are planning to implement include:

1) **Laborer Table**: contains the information about all the workers/laborers of the city and contains following attributes.

   Laborer(lab_id, Name, address, dist_id, Wage Type (Hourly / Weekly))

2) **Supervisor Table:** contains the information about all the supervisors and it has the following attribute fields.

   Supervisor (sup_id, Name, Address, dist_id)

3) **District Table :** contains the mapping of dist_id to the names of districts and contains the following attributes.

   District (dist_id, Name)

4) **Hourly Wage Rate:** specifies the hourly wage rate for a particular work type.

   Hourly Wage Rate (Work type, Wage rate)

5) **Daily Wage Rate**: specifies the daily wage rate (per day) for a particular work type and it will be useful for weekly wage earners.

   Daily Wage Rate (Work type, Wage rate)

6) **Work and Wage Table :** After this we need to show which workers are assigned to which Supervisor, this table will help us to relationship between supervisor and worker and It Contains the following attributes:

   Central Table (sup_id, lab_id, Wage Type (Hourly / Weekly))

7) **Hourly Earning :** This table is specifically designed for laborers earning hourly wages and it will be updated on a daily basis, it contains the following attributes:

Hourly Earning (lab_id, Wage Type, Hours worked, daily earnings (hours worked * wage rate))

8) **Weekly Earning:** This table is specifically designed for laborers earning weekly wages and it will be updated on a weekly basis, it contains the following attributes:

   Weekly Earning (lab_id, Work Type, Monday, Tuesday, Wednesday, Thursday,Friday, Saturday, Total Earning)

   Here, the attributes (Monday, Tuesday, Wednesday, …) represent presence/absence (boolean domain) of the laborer on that particular day and total earnings are calculated on the basis of number of days present in a week multiplied by wage rate of that work type.

[2]. Document the Requirements Collection/Fact-Finding Phase

2.1 Background Reading/s

- **H&M Group WMS (Wage Management System):** This management system has its working mechanism mainly oriented to achieve important objectives in the areas like proper and correct payment of wages, internal development of human resources, A progressive and coherent pay system, proper mechanisms of workers' involvement. These 4 parameters can be also used to measure performance metrics of this management system. Other than this, there are also some requirements needed for efficient functioning of this system like strict & clear definition of policies & procedures, responsible person department, communication & feedback from various departments using this system.

- **NALCO (A Govt. of India Enterprise):** This system helps in digitization of most of the transactions pertaining to contract labor engagement, thereby reducing human intervention and increasing transparency in the system. This system also incorporates the combined forms under the ease of compliance to maintain registers under various Labor laws & rules. This system maintains a comprehensive database of all contractors and contract laborers engaged at different Units/Offices of NALCO and will facilitate fulfillment of NALCO's responsibility as a sensible Principal Employer. This system has many features which include a mechanism to validate payment of minimum wages, other statutory dues like EPF and ESI, generate wage slips, employment cards and several statutory forms/ registers/ returns related to contract labor deployment as required. Compliance by contractors on payment of wages and other benefits to the contract laborers will be ensured through this management system.

References

https://hmgroup.com/wp-content/uploads/2020/10/Wage-Management-System-Guideline.pdf

https://nalcoindia.com/services-initiatives/contract-labour-management-system/

2.2 Interviews

(1)

Interviewers: 1) Vansh Parikh 2) Fenil Dalwala

Date: 26/9/2022, Time: 14:30
Duration: 45 minutes, Place: DAIICT

Interviewee Role - Laborer(Construction), Name: Suresh Sharma

Purpose of Interview:
Preliminary meeting to identify problems and requirements regarding Wage & Labor Distribution in the City

Questions:

1. **Which Skills do you possess?**

➢ I have skills of working in areas like mining, construction work and can also do tailoring & washingman work.

2. **Do you own a smartphone and do you know how to use it?**

➢ Yes, I have a smartphone & also have basic knowledge of its features & working.

3. **Have you ever been paid less wages by your district supervisor than the decided amount?**

➢ No, there has been no such experience till now but sometimes we are required to work overtime without being paid for that.

4. **Have you ever complained about something to your supervisor and was the required action taken?**

➢ No, once I complained on the behalf of all laborers about the conditions of the stay which was offered to the laborers during a construction project but it was not listened to.

5. **Do you feel the upcoming wage management system will solve the current problems of laborers?**

➢ Yes, I feel to some extent that it will provide us transparency about the system & a platform on which the workers will rely for the resolution of their complaints.

Summary:

From this person's interview we got basic information about his occupation also and wage type, in addition we able to find the problem which he was facing about stay and he didn't get any response but by our wage management system we can resolve this kind of problem in better way by our feedback system and simultaneously he will get early response from administration because it is online system.

(2)

Interviewers: 1) Vansh Parikh 2) Fenil Dalwala

Date: 25/9/2022, Time: 14:30
Duration: 45 minutes, Place: Google Meet

Interviewee Role - District Supervisor, Name: Ashok Shukla

Purpose of Interview:

Preliminary meeting to identify problems and requirements regarding Wage & Labor Distribution in the City

Questions:

1. **Approximately how much workforce is there in your district?**

➢ There are approximately 2000 workers under my supervision.

2. **Do you sometimes face problems in managing such a large workforce?**

➢ Not always, but yes sometimes it becomes hectic to manage all the complaints/queries of the laborers.

3. **Do you feel that the government's policies & procedure of wage distribution are properly implemented at microscale?**

➢ Yes, at least in my district, I can say that we try our level best to efficiently implement them but sometimes it is possible that we might oversee it due to human errors.

4. **How do you keep track of the wage distribution among laborers and are you facing any problems?**

➢ We keep track of Wages distributed with the help of manual registers and it is a usual practice in all other districts also but yes, sometimes it is difficult to maintain the consistency of the registers.

5. **Do you feel the upcoming wage management system will solve the current management issues of district supervisors?**

➢ Yes, I feel that this Wage management system will take the city labor management in a positive direction and we will have a consistent database for wage distribution.

Summary:

By supervisor's interview, we can see that wage distribution policy is correctly implemented in his district however maintaining data of wage is become challenging for him and and updation of data in wages register is very hard for him, our wage distribution system can help him in this aspect and he is also hoping that this issue will be resolved by our system.

(3)

Interviewers: 1) Vansh Parikh 2) Fenil Dalwala

Date: 25/9/2022, Time: 14:30
Duration: 45 minutes, Place: Google Meet

Interviewee Role - City Administration Officer,
Name: Manoj Patel

Purpose of Interview:
Preliminary meeting to identify problems and requirements regarding Wage & Labor Distribution in the City

Questions:

1. **As the city is currently undergoing development, how do you plan to increase the strength of the labor workforce for the city?**

➢ Yes, sure as the city will be developing, we will be needing a large workforce for consistent development, new employment opportunities will be created and workers will be recruited on the basis of their skills and requirements for the city at that time.

2. **What are your thoughts on the upcoming wage management system?**

➢ Like as we discussed the increasing workforce will also require an efficient labor & wage management system for the smooth development process of the city. We, at the central administration of the city, will be keeping track of works currently going on in all the districts of the city and will also be monitoring the wage distribution among the laborers as well as through this system, we will try our best to hear the problems of each and every laborer of the city.

Summary:
From this interview, we got to know that the city administration is also looking for a system which can give them centralized control over the city labor where each & every

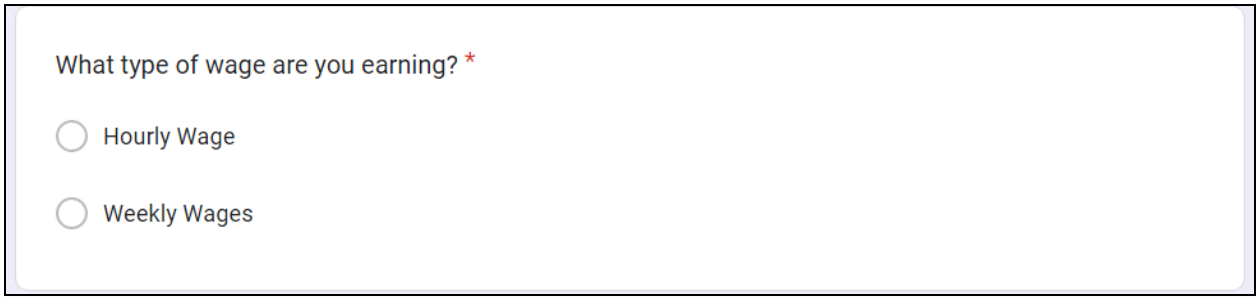
laborer is connected in a single process & doesn't become a victim because of faults in managing a large workforce.

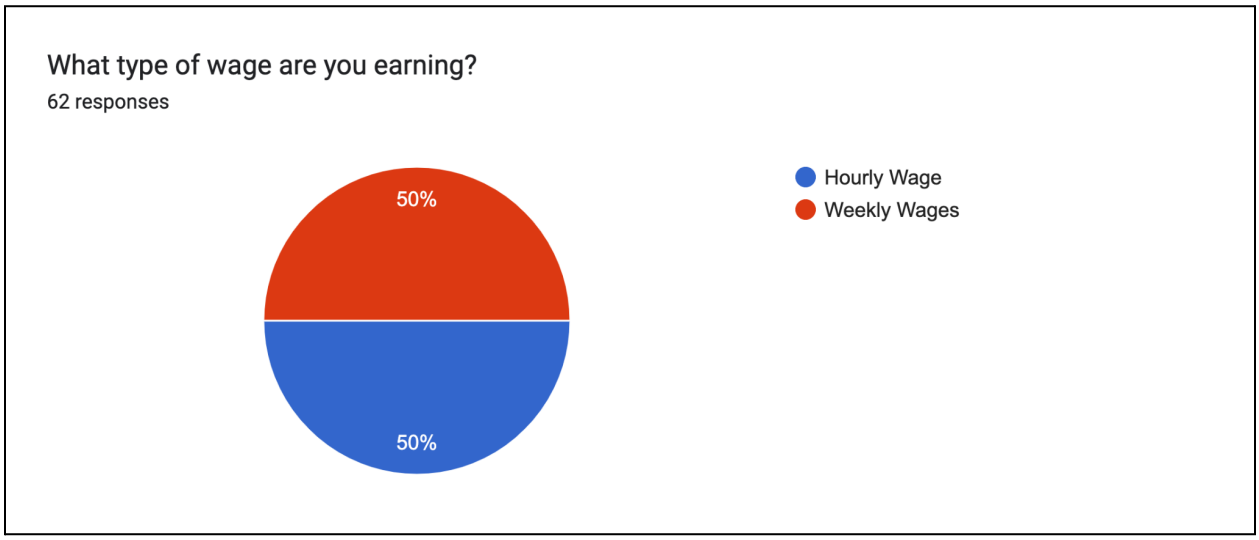
Combined Requirements gathered from interviews:

- A well-functioning Database management system is required to maintain and update all the information about laborers, Supervisors, Administrators, etc.
- Structure and functionalities of the system should be designed in such a way that it ensures the optimal performance of the system.
- The different user interfaces are required for the different user Entities(laborers, Supervisors, Administrators) so that the particular user can access only the corresponding data and functionalities.
- The interface of the database should be clean and without any redundant data.

2.3 Questionnaire

1. Collecting Data of Wage Type





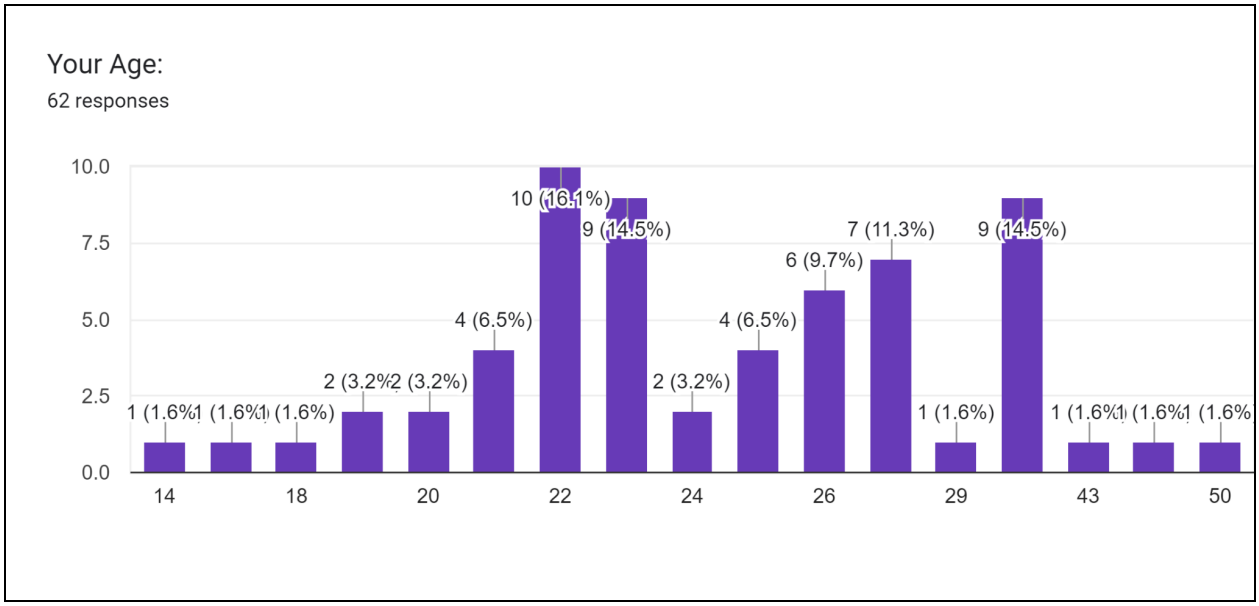
Intent of the Question:
To calculate the number of laborers earning a particular wage type.

Observation:
Here, we observe that hourly wage and weekly wage laborer numbers are almost the same.

2. Age of Laborers

Your Age: *

Short answer text



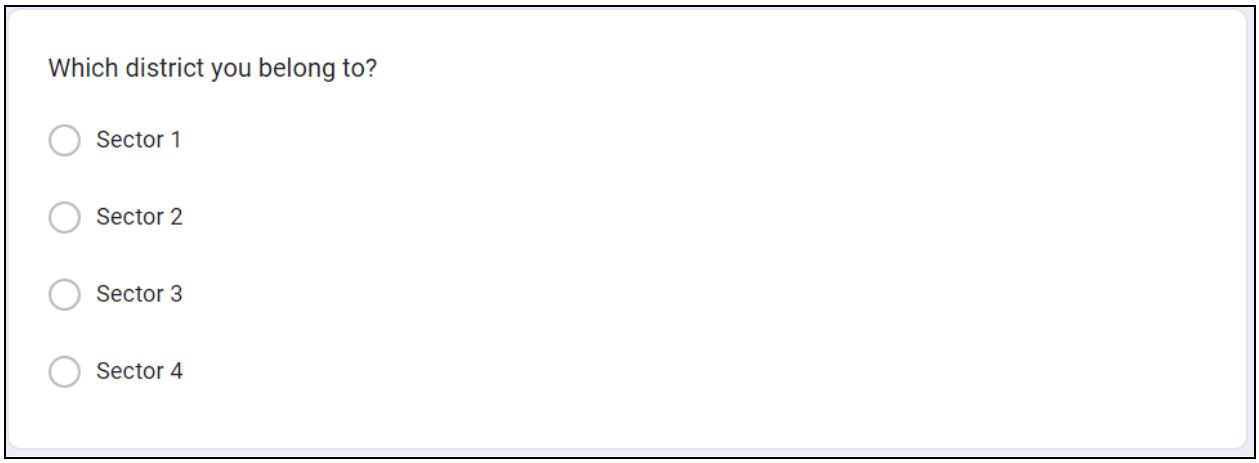
Intent of the Question:

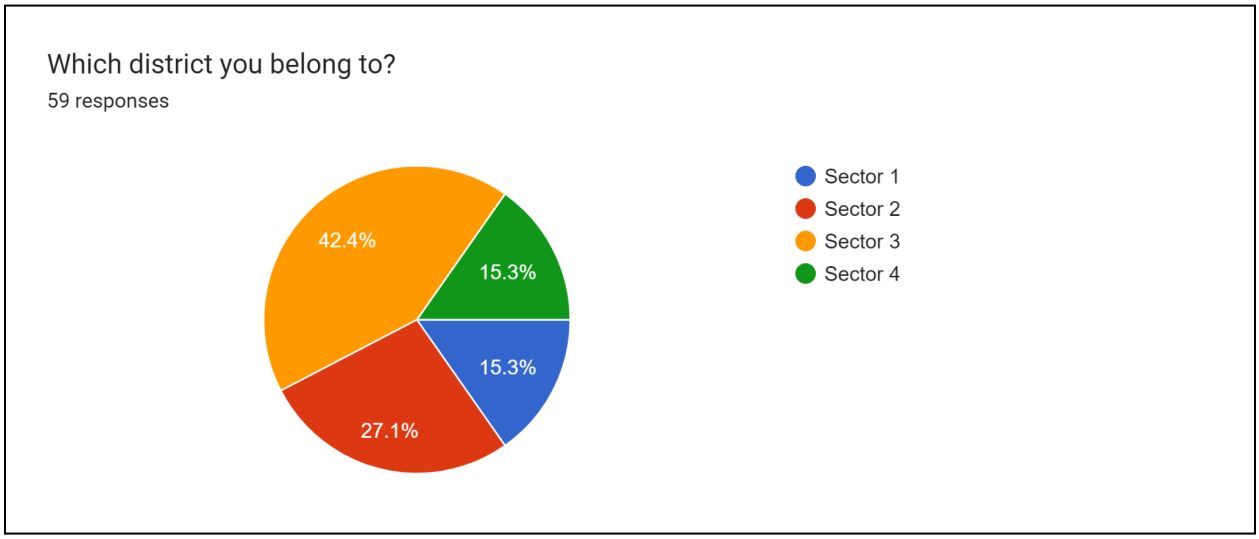
To find age of laborers

Observation:

We can see that the average age of laborers lie between 22-24 yrs and most of the laborers have an age between 20-30 yrs.

3. Number of Laborers in a district





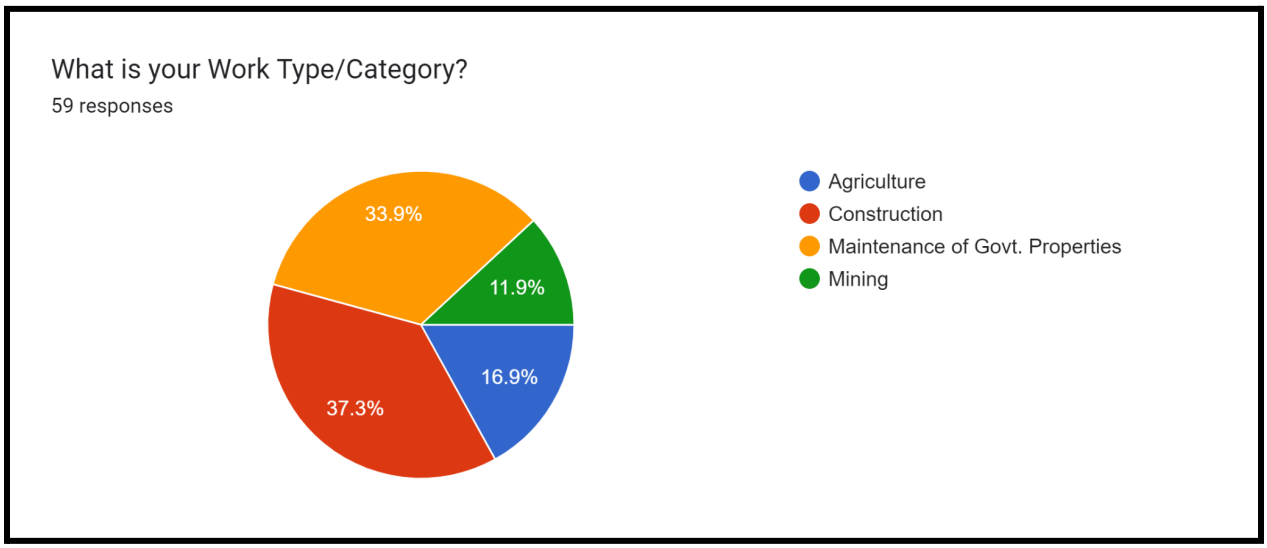
Intent of the Question:
To find out laborer shares of different districts

Observation:
Here, we can see that section 3 dominates the most with 42.4% of total laborers and section 1 and section 4 have least among all with 15.3% employees laborers.

4. Information about Work Type of Laborers

What is your Work Type/Category? *

- ○ Agriculture
- ○ Construction
- ○ Maintenance of Govt. Properties
- ○ Mining



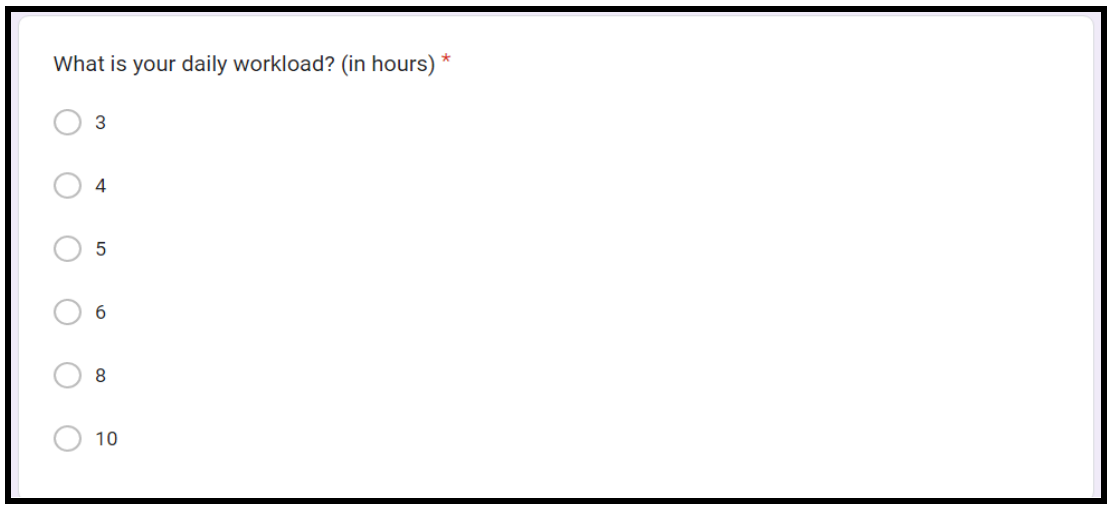
Intent of the Question:

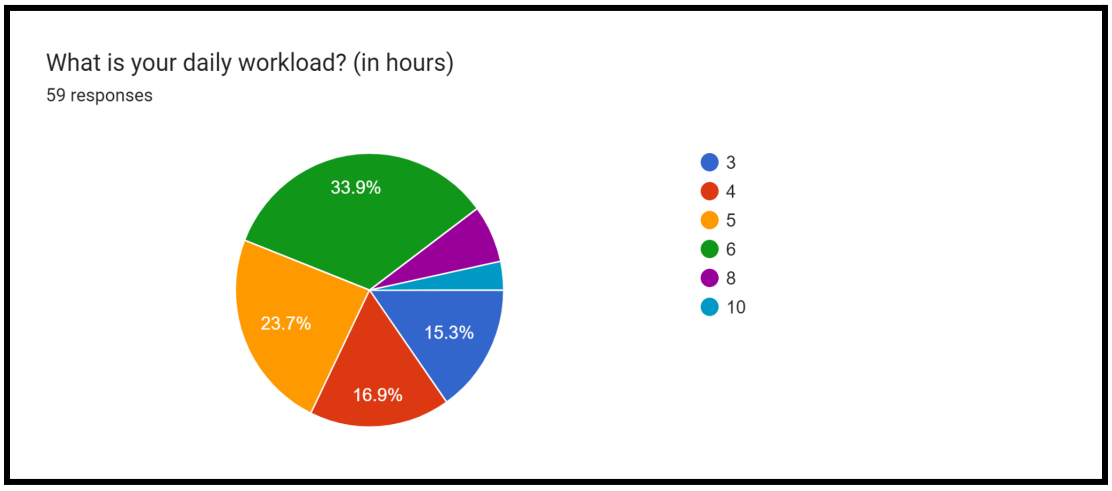
To find out number of laborers doing a particular type of work

Observations:

We can see that nearly $\frac{1}{3}rd$ of employees are associated with construction occupation, mining provides 11.3% employment which is least among all and other than this 34% employees are involved in Maintenance of Government Properties.

5. Daily Hours of Work





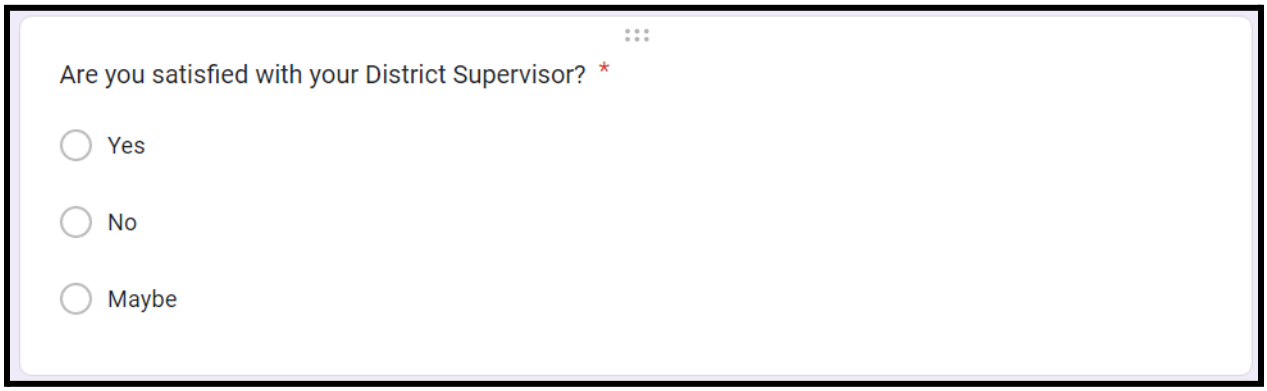
Intent of the Question:

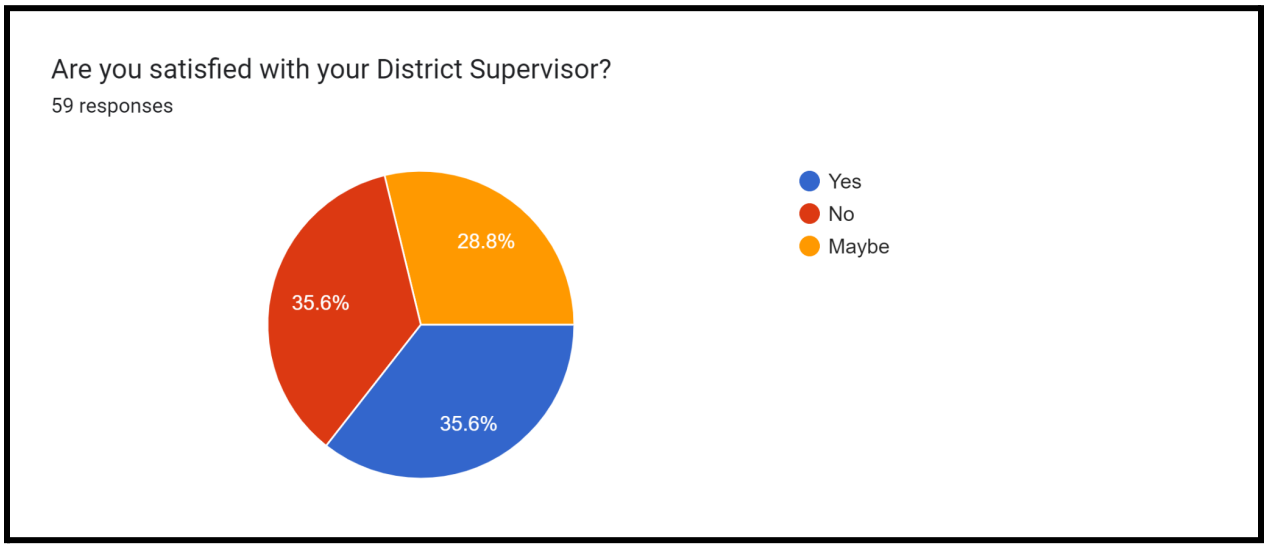
This question will give us information that how much time a person is giving to its work daily which will help us for hourly as well as weekly wage types employees,As for hourly wage laborers we can calculate their wage by their daily working hours and this information helps to manage daily attendance of weekly wage laborers.

Observation:

We can see that total average hours a laborer is working is 6 hours and 34% laborers work 6 hours a day, other than this there were very few laborers who work more than 8 hours a day, 24.2% laborers works 5 hours and following that 16.1% laborers work 4 hours a day.

6. Satisfaction of laborers with the supervisors

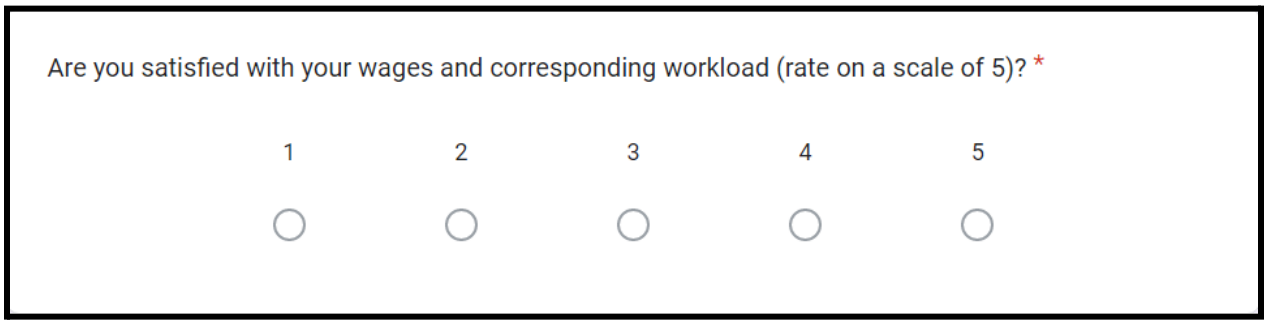


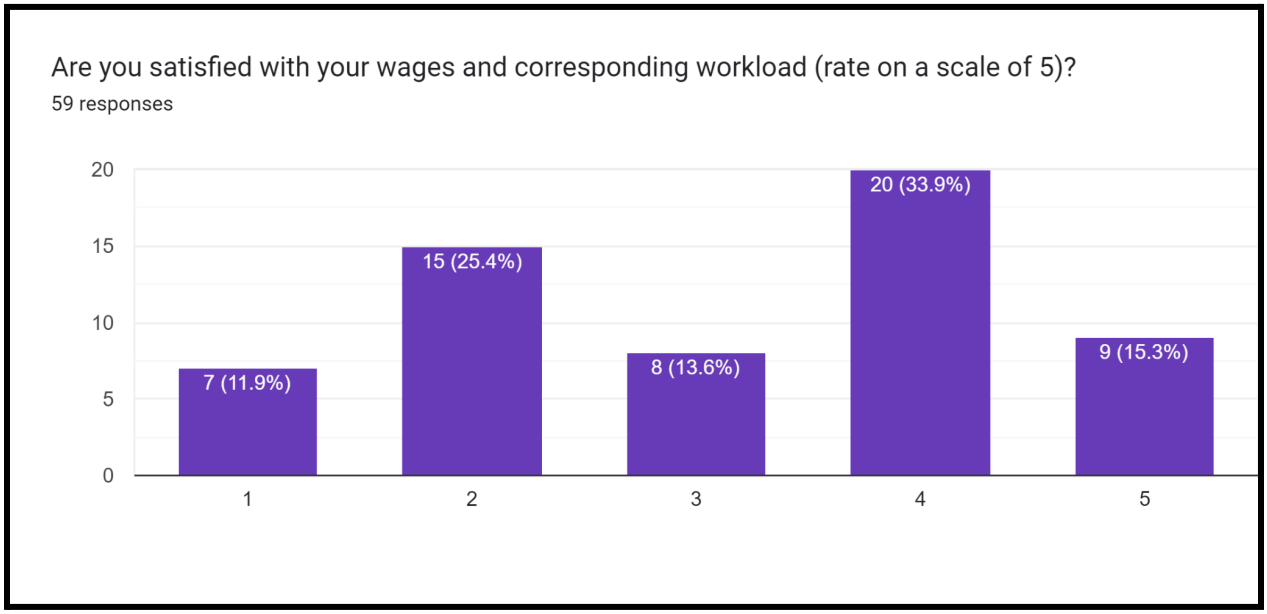


Intent of the Question:
This Question will allow our system to measure satisfaction of laborers regarding their District Supervisors which will also influence district supervisors activities and regulate their actions.

Observation:
We observe that over 35% laborers are satisfied with their supervisors but on the other hand 28% are not,in addition to this over 28% employees are not sure about their opinion so we can consider their maybe as yes or no but by optimistic view we can say that most of them are satisfied with supervisors however there may be some laborers facing little issues as well.

7. Satisfaction of Laborers with Wages & Workload





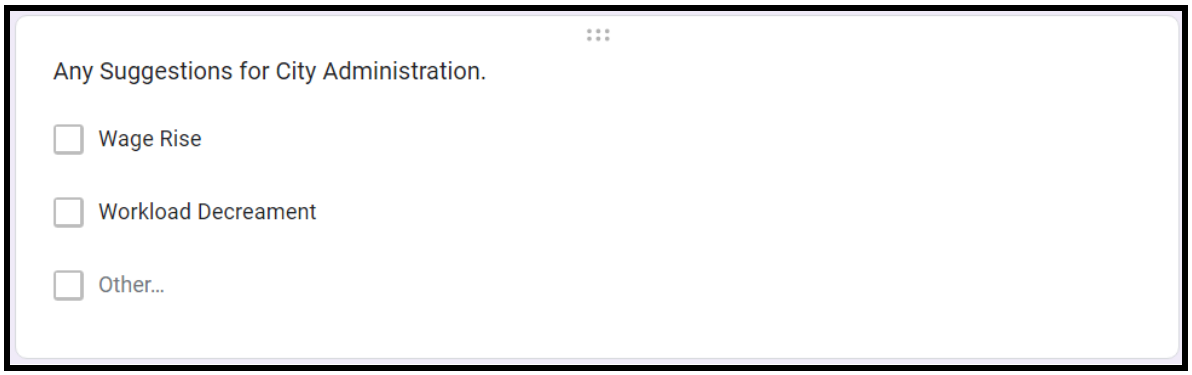
Intend of the Question:

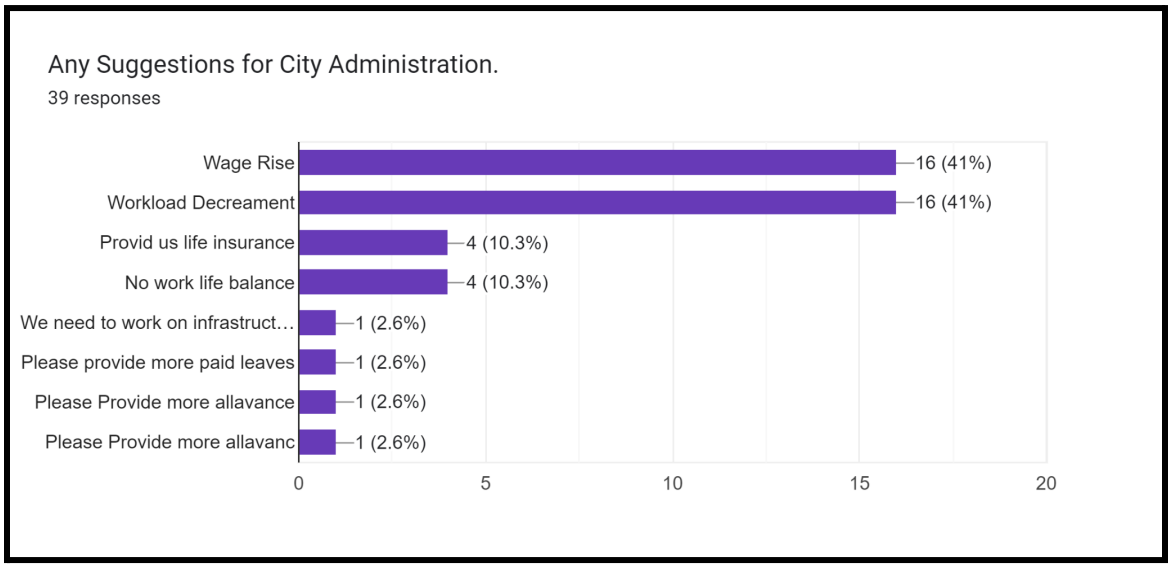
This Question allows laborers to rate their satisfaction about their wage as well as workload.

Observation:

This Question got mixed reviews. We can say that each satisfaction rate almost got the same votes but most votes went to scales 2 and 4 and the extremes got the least votes.

8. Laborer's feedback





Intent of the Question:

This Question will help with suggestions from the laborer's end and this will help our system to be better and it will also draw our attention to any genuine problems related to laborers or a general demand of laborers.

Observations:

We can see that nearly 40-40 percent votes go for wage rise and workload decrement which is general suggestions other than this laborers suggest new suggestions like getting life insurance or maintaining work life balance which are also debatably good points to make, furthermore there are suggestions for increasing paid leaves and allowances.

2.4 Observation/s

System: NALCO India

Observations by: 1) Vansh Parikh, 2) Fenil Dalwala

Date: October 1, 2022

Time taken: 1 hr

Observations and Combined Requirements:

- **Responsible Person's Details:** We observe that the details of the Responsible person must be clear in the procedural document about his work area, his responsibilities, his restrictions and his working activities. As in our case we can say that Supervisor's role details must be descriptive in the terms of his District and workers ids other than that how much his decisions will affect his district or other's district should also be mentioned in the document.

- **Policy and Procedure:** This will be the most important and critical aspect of any management system, its policies and procedures. On these, the management systems fundamentals are to be set, that's why they must be well defined. For our management system, all policies regarding wage distribution, government policies on laborers and district supervisors should be given in a structured way so that one can understand it properly. All the responsible persons or departments should have authority to enforce the policy and procedures

- **Communication and feedback:** Communication in any system will be the pillar of resemblance and feedback will assure integrity of a system

hence both will be unignorable parts of any management system. Communication channels should be used to increase awareness and knowledge on policies and procedures. There should be feedback mechanisms to understand and verify workers' awareness level. In our management system, we can allow laborers to communicate with their supervisors, so that one can give any kind of suggestion or valid feedback of new policies.

- **Monitoring:** It is how you check whether your procedures are being followed and working well and whether you're compliant with the law and your standards. This could involve a time-to-time age verification process and monitoring of working hours and pay of workers.

- **Required Technical Specifications:**
    - Efficient and optimal storage of all data in the system.
    - Real-time update of all the information stored in the database system, maintaining consistency of the database.
    - Efficient services related to the search, insert, delete, and update operations.
    - Well-designed to withstand traffic up to a certain level and should provide a quality experience to the users.
    - Concurrent access to the database system.
    - Security and integrity of the information stored in the database system.
    - Stable Internet connection

[3]. Fact-Finding Chart

Objective	Technique	Subject(s)	Time Commitment
To get the background of the Wage Management System and format of the SRS	Background Reading	Website, Book	1 Day
To gain an understanding of the roles of laborers	Questionnaire	Laborer	2 hours
To find out the roles of district supervisor	Interview	District Supervisor	1.5 hours
To gain an understanding of the roles of city administration	Interview	City Administration Officers	45 min
To determine the difficulty face by the laborers	Interview	Laborer	1 hour
To gain an understanding of the real world Wage Management database	Observation	Websites of H&M WMS, NALCO	1 hour

[4]. List Requirements

- The database should be consistent and without any redundant data.

- A well-functioning Database is required to maintain and update all the information about stored data.
- Real-time update of the information stored in the database system.
- The integrity in different system parts should be maintained in real-time.
- The database should be updated, and integrity should be maintained in real-time. Also, a backup should be provided for our database in case of inconsistency/crash.
- System structure and functions should be designed in such a way that it ensures the fast performance of the system.
- The different user interfaces are required for the other user entities (Laborers, Supervisors, Administrators) so that the particular user can access only the corresponding data and functionalities.
- Participants should be able to see the comments and feedback on their submissions given by judges.
- The general visitor should be able to see the dashboard & requested information if it is accessible by him.
- The Supervisors should be able to manage the activities of a particular cluster of laborers or we can say that laborers working under them.
- The Supervisors should be able to access all the details of the laborers.
- The laborers should be able to see the status of their applications or requests.
- The laborers should be able to give feedback and suggestions about their wages, working conditions & behavior of supervisors.
- There should be sufficient communication between the laborers and supervisors/administrators to resolve their problems or queries.
- City Administrators should be able to access all the information about laborers as well as supervisors and should be able to grant access or take back back access from anyone whenever they want.
- Laborers should be able to see checking his/her profile, working & wage history.

- The interface should be smooth and user friendly without any redundant data.
- There should be a feedback system which will encourage laborers' suggestions.

[5]. User Categories and Privileges

→ List of user categories and their roles

1) **Laborer:** The role of a laborer is offered to the end user & mainly for whom the management system is created. Laborer can perform various tasks like checking his/her profile, working & wage history, upcoming projects to apply for & wages offered in that.

2) **Supervisor:** The role of a supervisor is offered to an intermediate person between government & laborers, his work is to manage the activities of a particular cluster of laborers.

3) **City Administration:** This role is offered to officers working in the city administration office which is responsible for all activities related to labor in all the districts of the city, they can access all the data as well as can assign or take back any role from others.

4) **General Visitor:** This role is assigned to all the general public under the Right to Information Act (2005) which gives the permission to access required data which can be demanded by the citizens.

5) **Database System Manager:** The role of the database system manager is to look after the technical issues faced by the users, control the traffic over the database system and regularly take the backup of data stored in

the database system, maintaining the consistency and integrity of the database.

5.1 List of privileges/functions that can be accessed by different user classes:

1) Search:

- General Visitors
- Laborers
- Supervisors
- Administrator
- DB Manager

2) Insert:

- Supervisors
- Administrator
- DB Manager

3) Delete:

- Supervisors
- Administrator
- DB Manager

4) Update:

- Supervisors
- Administrator
- DB Manager

[6]. Assumptions

- The users of this database system are required to have all the requisite hardware and software to operate this application.
- Furthermore, it is requested that the database's data is updated in real-time.
- The database system's users are presumed to have reliable internet connections.
- The database is consistent.
- The database maintains the data's integrity all the time.
- In rare circumstances, the users will have access to substitute resources.

[7]. Business Constraints

- The amount of software and hardware required for the database system is limited.
- The computing power available for the database system is also limited.
- The capacity of the database is limited.
- The number of users accessing the database simultaneously is also limited.

II Noun Analysis

1. All Extracted Nouns & Verbs from Problem Description

Nouns	**Verbs**
Laborers	Weekly Earnings
Supervisors	Hourly Earnings
Administrators	Management
Visitors	payment
Workers	need
Wages	updated
Manager	Calculated
City	specifies
District	relate
Name	earning
Age	assigned
Address	according
User	Lived in
Password	
Presence	
Absence	
Log in	
Sign up	

2. Accepted Noun and verb list

Candidate Entity Set	Candidate attribute Set	Candidate Relation Set
Laborer	(lab_id, Name, address, dist_id, Wage Type (Hourly / Weekly))	Supervisor,District,Weekly Wage,Hourly Wage
Supervisor	(sup_id, Name, Address, dist_id)	Work Wage rate
District	(dist_id, Name)	
General Visitors	(Name,age,address)	
City Administrator officer		

3. Rejection Noun and Verb List

None	Reject Reason
Workers	Because it will be redundant and same as Laborers
Name	Name will be common attribute of all of our entities so don't need to separately create a new candidate key
City	Our System will work on only one city and it will manage different districts of the city so for our case city is not a candidate instead it is like a whole area of the system.
Manager	We Declare Supervisor as a candidate so we don't need to declare another term named Manager which we rise in redundancy.
Wage	We don't consider age as none, instead of that we can use it in relationship set.

III Identify Relationship types

1. Entity vs. Attribute

Entity	Attribute
User	Age, DOB, Password, Contact_Info, Email_Id, Name
Laborer	Dist_id, lab_id
Supervisor	Sup_id, dist_id
City Administration Officer	Admin_id
District	Dist_id, name
Daily Wage Laborer	Date, hours_worked, work_type
Weekly Wage Laborer	Work_type, Monday,Tuesday, Wednesday, Thursday, Friday, Saturday, Total Present days
General Visitor	date

2. **Entity vs. Relationships**

Entity	Relationships
Laborer, Supervisor	Works for
Supervisor,District	Manages
Laborer, District	Lives in
Daily Wage Laborers, Wage Rate	Daily Earnings
Weekly Wage Laborers, Wage Rate	Weekly Earnings

3. **Binary vs. Ternary Relationships:**

All relations between all the entities are binary in the ER diagram.

Works for(relationship between Laborer & Supervisor)

- ☐ This is a many-to-one relationship.

Manages(relationship between Supervisor & District)

- ☐ This is a one-to-one relationship.

Lives in(relationship between Laborer & District)

- ☐ This is a one-to-many relationship.

Daily Earnings(relationship between Daily Wage laborers & Wage rate)

- ☐ This is a many-to-one relationship.

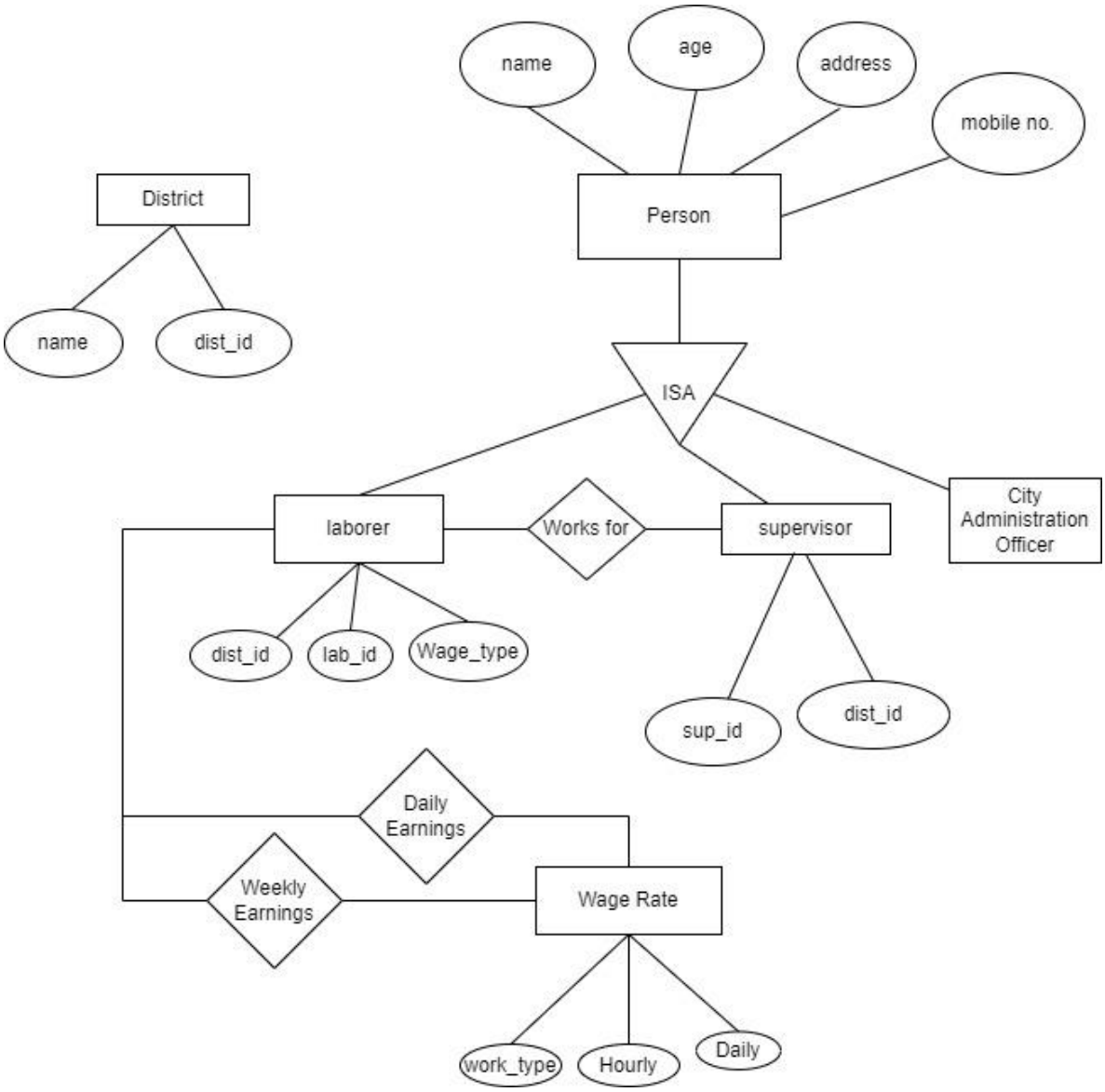
Weekly Earnings(relationship between Weekly Wage laborers & Wage rate)

- ☐ This is a many-to-one relationship.

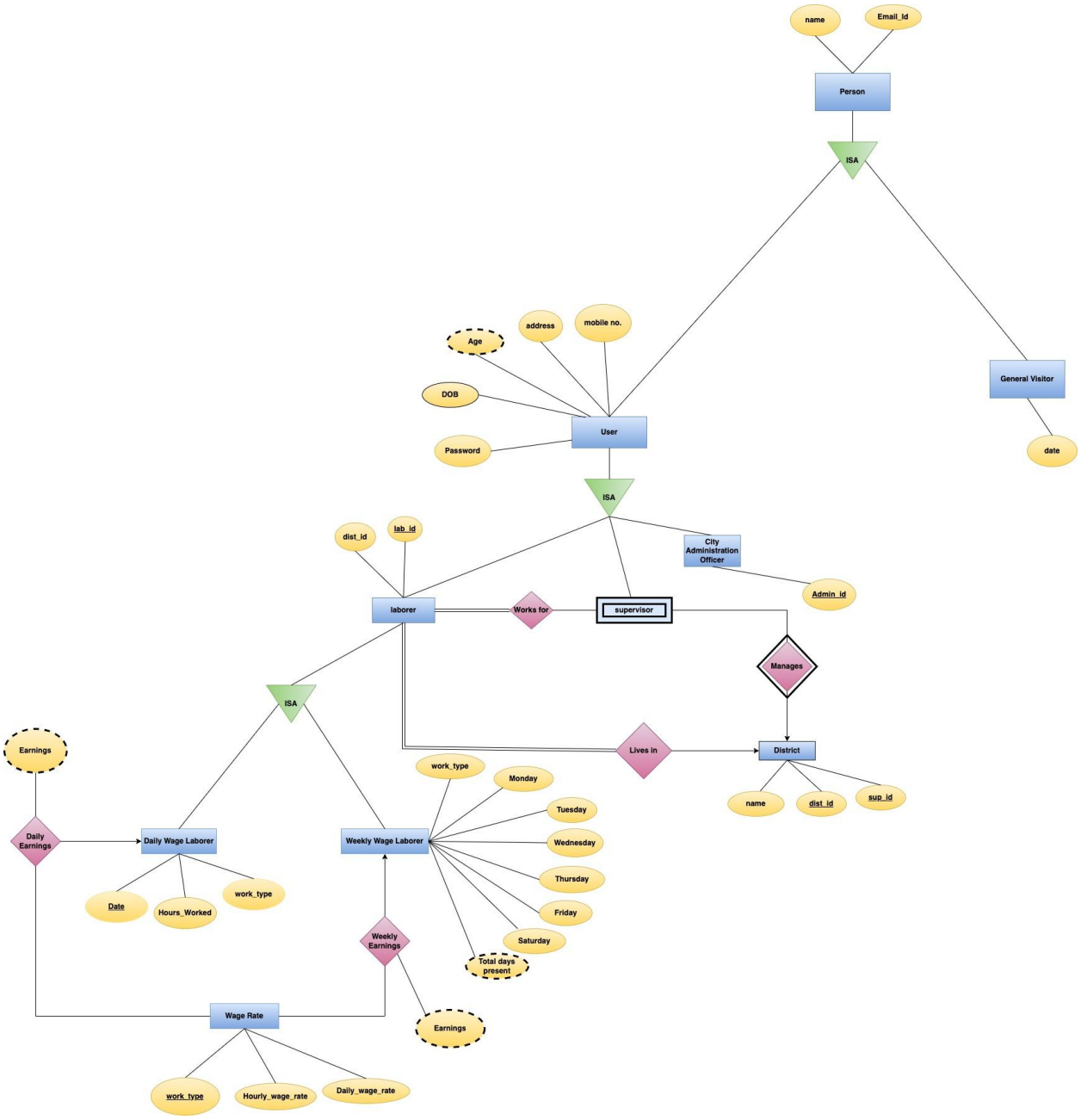
3) Analyze ERD for any other missing information.

- All entities are the same as before in the latest ER diagram.
- We added a **derived** attribute(Earnings, Age, Total Days Present).
- Instead of extra attributes we generalized that attribute and removed redundancy.
- We added a weak entity relationship between two relations and defined partial and identifying keys for this relation.
- Also we add ISA **specialization** from regular users to laborers, supervisors, and city administrator officers.
- Administrator has one extra attribute from the user entity such as Admin_id.
- Similarly the Laborer entity has a lab_id attribute more than the User entity.
- The ER diagram also includes various **weak entity** such as supervisor which depends on the district.

4. ER Diagram v1



5. ER Diagram v2



IV Conversion of Final ER-Diagram to Relational Model

Relations:

laborer(lab_id, name,age, address, mobile number, email_id, password, dist_id, sup_id);

Daily_wage_laborer(lab_id, date, hours_worked, work_type, Earnings);

Weekly_wage_laborer(lab_id, week_id, monday, tuesday, wednesday,thursday, friday, saturday, total_present, work_type, Earnings);

Supervisor(sup_id, name, age, address, mobile number, email_id, password);

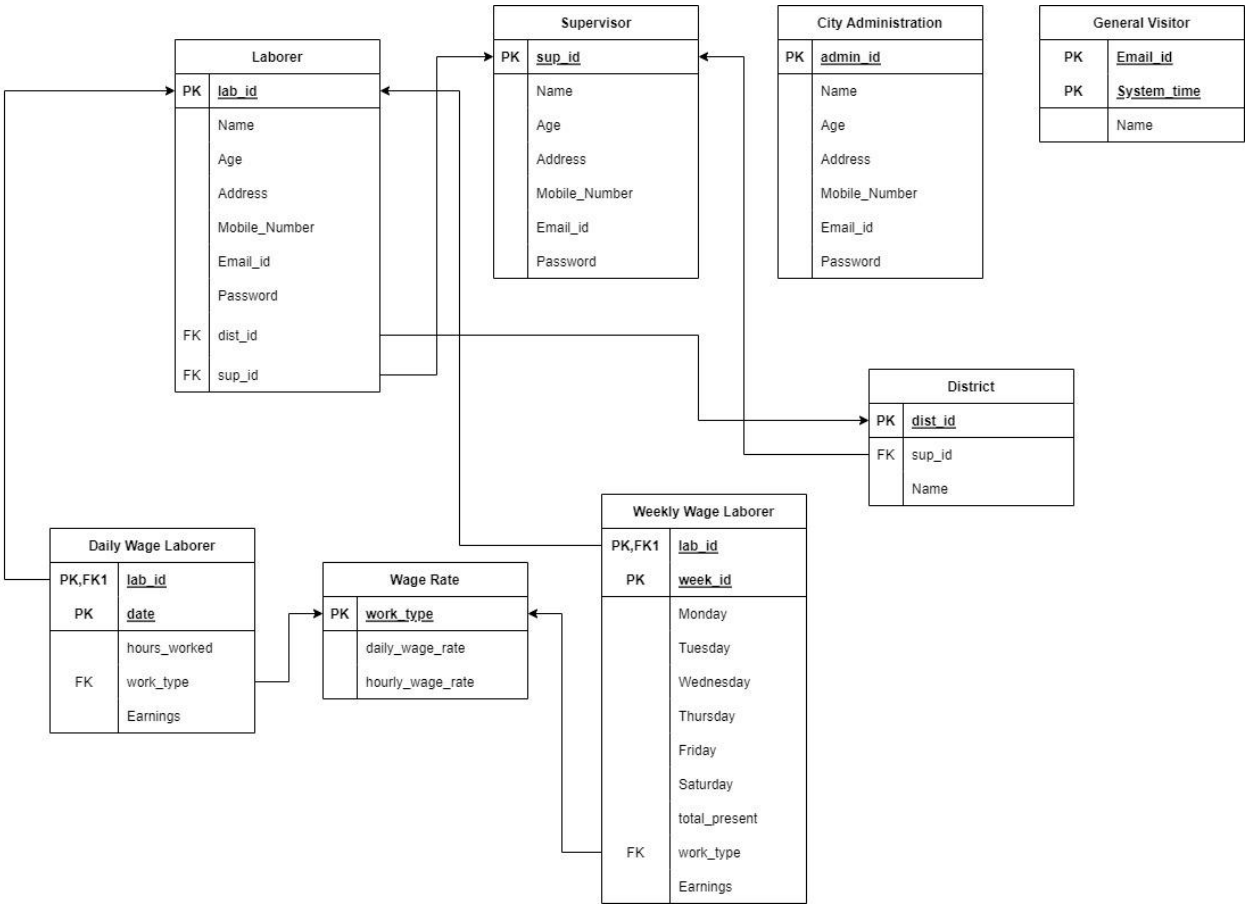
City_Administration(admin_id, name, age, address, mobile number, email, password);

District(dist_id, sup_id, name);

General Visitor (name, email_id, system_time);

Wage Rate (work_type, hourl`y_wage_rate, daily_wage_rate);

1. Relational Schema



2. DDL Scripts

1. Laborer:

```
CREATE TABLE IF NOT EXISTS laborer_db.laborer
(
Lab_id integer NOT NULL,
Name character varying,
Age integer,
Mobile_no integer,
Address character varying,
Email character varying,
password character varying,
Dist_id integer,
Sup_id integer

PRIMARY KEY (Lab_id),

FOREIGN KEY (Dist_id)
        REFERENCES laborer_db.District(Dist_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID

FOREIGN KEY (Sup_id)
        REFERENCES laborer_db.Supervisor(Sup_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID
```

```
);

ALTER TABLE IF EXISTS laborer_db.Laborer
    OWNER to postgres;
```

2. Supervisor:

```
CREATE TABLE IF NOT EXISTS laborer_db.Supervisor
(
Sup_id int NOT NULL,
Name character varying,
Age integer,
Mobile_no integer,
Address character varying,
Email character varying,
password character varying,

PRIMARY KEY (Sup_id),

);

ALTER TABLE IF EXISTS laborer_db.Supervisor
    OWNER to postgres;
```

3. District:

```
CREATE TABLE IF NOT EXISTS laborer_db.District
(
Dist_id integer NOT NULL,
Sup_id integer,
Name character varying,

PRIMARY KEY (Dist_id),
FOREIGN KEY (Sup_id)
        REFERENCES laborer_db.Supervisor(Sup_id) MATCH SIMPLE

        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID

);

ALTER TABLE IF EXISTS laborer_db.District
   OWNER to postgres;
```

4. Administrator:

```
CREATE TABLE IF NOT EXISTS laborer_db.City_Administration
(
Admin_id integer NOT NULL,
```

```
Name character varying,
Age integer,
Mobile_no integer,
Address character varying,
Email character varying,
password character varying,


PRIMARY KEY (Admin_id),

);

ALTER TABLE IF EXISTS laborer_db.City_Administration
    OWNER to postgres;
```

5. General Visitor:

```
CREATE TABLE IF NOT EXISTS laborer_db.General_Visitor
(
Name character varying,
Email character varying,
"System_Time" timestamp with time zone,


PRIMARY KEY (Email,System_Time),
);

ALTER TABLE IF EXISTS laborer_db.General_Visitor
```

```
    OWNER to postgres;
```

6. Wage Rate table:

```
CREATE TABLE IF NOT EXISTS laborer_db.Wage_Rate
(
Work_type character varying,
Hourly_Wage_Rate bigint,
Weekly_Wage_Rate bigint,

PRIMARY KEY (Work_Type)

);

ALTER TABLE IF EXISTS laborer_db.Wage_Rate
    OWNER to postgres;
```

7. Daily_wage_laborer:

```
CREATE TABLE IF NOT EXISTS laborer_db.Daily_wage_laborer
(
Lab_id integer NOT NULL,
date date NOT NULL,
```

```
Hours_worked integer NOT NULL,
Work_type character varying,
Earnings bigint,

PRIMARY KEY (Lab_id,date),
FOREIGN KEY (Lab_id)

        REFERENCES laborer_db.Laborer(Lab_id) MATCH SIMPLE

        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID

FOREIGN KEY (Work_type)

        REFERENCES laborer_db.Wage_Rate(work_type) MATCH SIMPLE

        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID
        );

ALTER TABLE IF EXISTS laborer_db.Daily_Wage_Laborer
   OWNER to postgres;
```

8. Weekly_wage_laborer:

```
CREATE TABLE IF NOT EXISTS laborer_db.weekly_wage_laborer
(
Lab_id integer NOT NULL,
```

```
week_id integer NOT NULL,
Monday boolean,
Tuesday boolean,
Wednesday boolean,
Thursday boolean,
Friday boolean,
Saturday boolean,
Total_present integer,
Work_type character varying,
Earnings bigint,

PRIMARY KEY (Lab_id,week_id),
FOREIGN KEY (Lab_id)

        REFERENCES laborer_db.Laborer(Lab_id) MATCH SIMPLE

        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID

FOREIGN KEY (work_type)

        REFERENCES laborer_db.Wage_Rate(work_type) MATCH SIMPLE

        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID
);
```

```
ALTER TABLE IF EXISTS laborer_db.Weekly_Wage_Laborer
    OWNER to postgres;
```

V Normalization and Schema Refinement

Original Relational Schema:

laborer(lab_id, name,age, address, mobile number, email_id, password, dist_id, sup_id);

Daily_wage_laborer(lab_id, date, hours_worked, work_type, Earnings);

Weekly_wage_laborer(lab_id, week_id, monday, tuesday, wednesday,thursday, friday, saturday, total_present, work_type, Earnings);

Supervisor(sup_id, name, age, address, mobile number, email_id, password);

City_Administration(admin_id, name, age, address, mobile number, email, password);

District(dist_id, sup_id, name);

General Visitor (name, email_id, system_time);

Wage Rate (work_type, hourly_wage_rate, daily_wage_rate);

Normalization & Schema Refinement:

1. **laborer(lab_id, name,age, address, mobile number, email_id, password, dist_id, sup_id)**

Functional Dependencies:
lab_id --> name
lab_id --> age
lab_id --> address
lab_id --> mobile number
lab_id --> email_id
lab_id --> password
lab_id --> dist_id
dist_id --> sup_id

Candidate Keys:
{(lab_id,dist_id)}

In this schema, every attribute is single-valued (scalar) making it already in 1NF.

Anomalies:
Insert – None.
Update – Updating this table will create problems sometimes as its primary key is being accessed as a foreign key in other tables.
Delete – Deleting from this table will create problems sometimes as its primary key is being accessed as a foreign key in other tables.

Partial Key Dependencies: None
2NF Redundancies: None

2NF requirements already satisfied.

Transitive Dependency: dist_id --> sup_id
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF tables are obtained after decomposition:

(<u>lab_id</u>,name,age,address,mobile_number,email_id,password,dist_id)

With FDs
lab_id --> dist_id,name,age,address,mobile_number,email_id,password

(<u>dist_id</u>, sup_id)
with FDs
dist_id --> sup_id

Since, there are no transitive dependencies remaining, the tables are in 3NF.

For a relation to be in BCNF,
a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, the relations are in BCNF as well.

2. **Daily_wage_laborer(lab_id, date, hours_worked, work_type, Earnings)**

Functional Dependencies:
lab_id,date --> hours_worked
lab_id,date --> work_type
lab_id,date --> Earnings
hours_worked,work_type --> Earnings

Candidate Key:
{(lab_id,date)}

In this schema, every attribute is single-valued (scalar) making it already in 1NF.

Anomalies:
Insert – None.
Update – Updating this table will create problems sometimes as its primary key is being accessed as a foreign key in other tables.
Delete – Deleting from this table will create problems sometimes as its primary key is being accessed as a foreign key in other tables.

Partial Key Dependencies: None
2NF Redundancies: None

2NF requirements already satisfied.

FD: **hours_worked,work_type --> Earnings**

The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).

The following 3NF tables are obtained after decomposition:

(lab_id,date,hours_worked,work_type)
with FDs
date,lab_id --> work_type,hours_worked

(hours_worked,work_type,Earnings)
with FDs
hours_worked,work_type --> Earnings

Since, there are no transitive dependencies remaining, the tables are in 3NF.

For a relation to be in BCNF,
a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, the relations are in BCNF as well.

3. **Weekly_wage_laborer(lab_id, week_id, monday, tuesday, wednesday,thursday, friday, saturday, total_present, work_type, Earnings);**

Functional Dependencies:
lab_id,week_id --> monday
lab_id,week_id --> tuesday
lab_id,week_id --> wednesday
lab_id,week_id --> thursday
lab_id,week_id --> friday
lab_id,week_id --> saturday

lab_id,week_id --> total_present
lab_id,week_id --> work_type
total_present,work_type --> Earnings
monday,tuesday,wednesday,thursday,friday,saturday → total_present

Candidate Keys:
{(lab_id,week_id)}

Anomalies:
Insert – None.
Update – Updating this table will create problems sometimes as its primary key is being accessed as a foreign key in other tables.
Delete – Deleting from this table will create problems sometimes as its primary key is being accessed as a foreign key in other tables.

In this schema, every attribute is single-valued (scalar) making it already in 1NF.

Partial Key Dependencies: None
2NF Redundancies: None

2NF requirements already satisfied.

FDs:

I. **monday,tuesday,wednesday,thursday,friday,saturday → total_present**
II. **total_present,work_type --> Earnings**

The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).

The following 3NF tables are obtained after decomposition:

R1(lab_id,week_id,monday,tuesday,wednesday,thursday,friday,saturday,total_present,work_type)
with FDs
lab_id,week_id --> work_type,monday,tuesday,wednesday,thursday,friday,saturday

R2(monday,tuesday,wednesday,thursday,friday,saturday,work_type,Earnings)
with FDs
monday,tuesday,wednesday,thursday,friday,saturday,work_type --> Earnings

R3(monday,tuesday,wednesday,thursday,friday,saturday,total_present)
With FDs
monday,tuesday,wednesday,thursday,friday,saturday → total_present

The relation R3 coming out of decomposition is a redundant table and will not be used during query creation.

4. **Supervisor (sup_id, name, age, address, mobile number, email_id, password);**

Functional Dependencies:
sup_id --> name
sup_id --> age
sup_id --> address
sup_id --> mobile_number
sup_id --> email_id
sup_id --> password

Candidate Keys:
{(sup_id)}

Anomalies:
Insert – None.
Update – None.
Delete – None.

In this schema, every attribute is single-valued (scalar) making it already in 1NF.

Partial Key Dependencies: None
2NF Redundancies: None

2NF requirements already satisfied.

Since there is no transitive dependency, it is also in 3NF.

Thus, our final relation remains the same.

For a relation to be in BCNF,
a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

5. **City_Administration(admin_id, name, age, address, mobile number, email, password);**

Functional Dependencies:
admin_id --> age
admin_id --> name
admin_id --> address
admin_id --> mobile_number
admin_id --> email
admin_id --> password

Candidate Keys:
{(admin_id)}

In this schema, every attribute is single-valued (scalar) making it already in 1NF.

Anomalies:
Insert – None.
Update – None.
Delete – None.

Partial Key Dependencies: None
2NF Redundancies: None

2NF requirements already satisfied.

Since there is no transitive dependency, it is also in 3NF.

Thus, our final relation remains the same.

For a relation to be in BCNF,

a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

6. **District (dist_id, sup_id, name)**

Functional Dependencies:
dist_id --> sup_id
dist_id --> name

Candidate Keys:
{(Dist_id)}

Anomalies:
Insert – None.
Update – Updating this table will create problems sometimes as its primary key is being accessed as a foreign key in other tables.
Delete – Deleting from this table will create problems sometimes as its primary key is being accessed as a foreign key in other tables.

In this schema, every attribute is single-valued (scalar) making it already in 1NF.

Partial Key Dependencies: None
2NF Redundancies: None

2NF requirements already satisfied.

Since there is no transitive dependency, it is also in 3NF.

Thus, our final relation remains the same.

For a relation to be in BCNF,
a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

Note:

This table serves the same functionality as the table R(dist_id,sup_id) which came as a result of decomposition of Laborer(1) during conversion to 3NF .

7. **General Visitor (name, email_id, system_time)**

Functional Dependencies:
email_id,system_time --> name

Candidate Keys:
{(email_id,system_time)}

Anomalies:
Insert – None.
Update – None.
Delete – None.

In this schema, every attribute is single-valued (scalar) making it already in 1NF.

Partial Key Dependencies: None
2NF Redundancies: None

2NF requirements already satisfied.

Since there is no transitive dependency, it is also in 3NF.

Thus, our final relation remains the same.

For a relation to be in BCNF,
a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

8. **Wage Rate (<u>work_type</u>, hourly_wage_rate, daily_wage_rate)**

Functional Dependencies:
work_type --> hourly_wage_rate
work_type --> daily_wage_rate

Candidate Keys:
{(Work_type)}

Anomalies:
Insert – None.
Update – None.
Delete – None.

In this schema, every attribute is single-valued (scalar) making it already in 1NF.

Partial Key Dependencies: None
2NF Redundancies: None

2NF requirements already satisfied.

Since there is no transitive dependency, it is also in 3NF.

Thus, our final relation remains the same.

For a relation to be in BCNF,
a. It should be in the third normal form (3NF).
b. For any dependency A → B, A must be a super key.

Hence, this relation is in BCNF as well.

1. **Normalized Relational Schema:**

laborer(lab_id, name,age, address, mobile number, email_id, password, dist_id, sup_id);

Daily_wage_laborer(lab_id, date, hours_worked, work_type);

Weekly_wage_laborer(lab_id, week_id, monday, tuesday, wednesday,thursday, friday, saturday, work_type);

Supervisor(sup_id, name, age, address, mobile number, email_id, password);

City_Administration(admin_id, name, age, address, mobile number, email, password);

District(dist_id, sup_id, name);

General Visitor (name, email_id, system_time);

Wage Rate (work_type, hourly_wage_rate, daily_wage_rate);

Daily_possible_earnings(hours_worked, work_type, earnings);

Weekly_possible_earnings(monday, tuesday, wednesday, thursday, friday, saturday, work_type, earnings);

VI SQL: Final DDL Scripts, Insert statements, 20 SQL Queries with Snapshots of output of each query

1. Modified DDL Scripts

wage_rate

```
CREATE TABLE public.wage_rate
(
    work_type character varying NOT NULL,
    daily_wage_rate integer,
    hourly_wage_rate integer,
    PRIMARY KEY (work_type)
);

ALTER TABLE IF EXISTS public.wage_rate
    OWNER to postgres;
```

City_administration

```
CREATE TABLE public.city_administration
(
    admin_id bigint NOT NULL,
    name character varying,
    age integer,
    address character varying,
    mobile_number bigint,
    email_id character varying,
    password character varying,
    PRIMARY KEY (admin_id)
);

ALTER TABLE IF EXISTS public.city_administration
    OWNER to postgres;
```

Supervisor

```
CREATE TABLE public.supervisor
(
    sup_id bigint NOT NULL,
    name character varying,
    age integer,
    address character varying,
    mobile_number bigint,
    email_id character varying,
    password character varying,
    PRIMARY KEY (sup_id)
);

ALTER TABLE IF EXISTS public.supervisor
    OWNER to postgres;
```

District

```
CREATE TABLE public.district
(
    dist_id bigint NOT NULL,
    sup_id bigint,
    name character varying,
    PRIMARY KEY (dist_id),
    FOREIGN KEY (sup_id)
        REFERENCES public.supervisor (sup_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

ALTER TABLE IF EXISTS public.district
    OWNER to postgres;
```

Laborer

```
CREATE TABLE public.laborer
(
    lab_id bigint NOT NULL,
    name character varying,
    age integer,
    address character varying,
    mobile_number bigint,
    email_id character varying,
    password character varying,
    dist_id bigint,
    PRIMARY KEY (lab_id),
    FOREIGN KEY (dist_id)
        REFERENCES public.district (dist_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

ALTER TABLE IF EXISTS public.laborer
    OWNER to postgres;
```

Daily_wage_laborer

```
CREATE TABLE public.daily_wage_laborer
(
    lab_id bigint NOT NULL,
    date date NOT NULL,
    hours_worked integer,
    work_type character varying,
    PRIMARY KEY (lab_id, date),
    FOREIGN KEY (lab_id)
        REFERENCES public.laborer (lab_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    FOREIGN KEY (work_type)
        REFERENCES public.wage_rate (work_type) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

ALTER TABLE IF EXISTS public.daily_wage_laborer
    OWNER to postgres;
```

Weekly_wage_laborer

```
CREATE TABLE public.weekly_wage_laborer
(
    lab_id bigint NOT NULL,
    week_id bigint NOT NULL,
    monday boolean,
    tuesday boolean,
    wednesday boolean,
    thursday boolean,
    friday boolean,
    saturday boolean,
    work_type character varying,
    PRIMARY KEY (lab_id, week_id),
    FOREIGN KEY (lab_id)
        REFERENCES public.laborer (lab_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    FOREIGN KEY (work_type)
        REFERENCES public.wage_rate (work_type) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

ALTER TABLE IF EXISTS public.weekly_wage_laborer
    OWNER to postgres;
```

General_visitor

```
CREATE TABLE public.general_visitor
(
    email character varying NOT NULL,
    system_time time without time zone NOT NULL,
    name character varying,
    PRIMARY KEY (email, system_time)
);

ALTER TABLE IF EXISTS public.general_visitor
    OWNER to postgres;
```

Daily_possible_earnings

```
CREATE TABLE public.daily_possible_earnings
(
    hours_worked integer NOT NULL,
    work_type character varying NOT NULL,
    earnings integer,
    PRIMARY KEY (hours_worked, work_type)
);

ALTER TABLE IF EXISTS public.daily_possible_earnings
    OWNER to postgres;
```

Weekly_possible_earnings

```
CREATE TABLE public.weekly_possible_earnings
(
    monday boolean NOT NULL,
    tuesday boolean NOT NULL,
    wednesday boolean NOT NULL,
    thursday boolean NOT NULL,
```

```
    friday boolean NOT NULL,
    saturday boolean NOT NULL,
    work_type character varying NOT NULL,
    earnings integer,
    PRIMARY KEY (monday, tuesday, wednesday, thursday, friday, saturday, work_type)
);

ALTER TABLE IF EXISTS public.weekly_possible_earnings
    OWNER to postgres;

    OWNER to postgres;
```

Queries

2. Simple Queries

1. Find all data of admin from the general administration table whose age is 33.

Select name,email_id,age
From labor_db.city_administration
Where age = 33;

Query | Query History

```
1  select name,email_id,age
2  from labor_db.city_administration
3  where age = 33;
```

Data Output | Messages | Notifications

	name character varying	email_id character varying	age integer
1	Cordell Hansel	chansel0@psu.edu	33
2	Storm Suttaby	ssuttabyg@ucsd.edu	33
3	Theda Bubbins	tbubbinsv@1und1.de	33
4	Zola Beames	zbeames2j@eventbrite.com	33

2. count of laborers whose work type is “construction” in daily_wage_laborer.

Select count(lab_id)
From daily_wage_laborer
Where work_type = ‘Construction’;

Query Query History

```
1
2  select count(lab_id)
3  from daily_wage_laborer
4  where work_type = 'Construction';
5
6
```

Data Output Messages Notifications

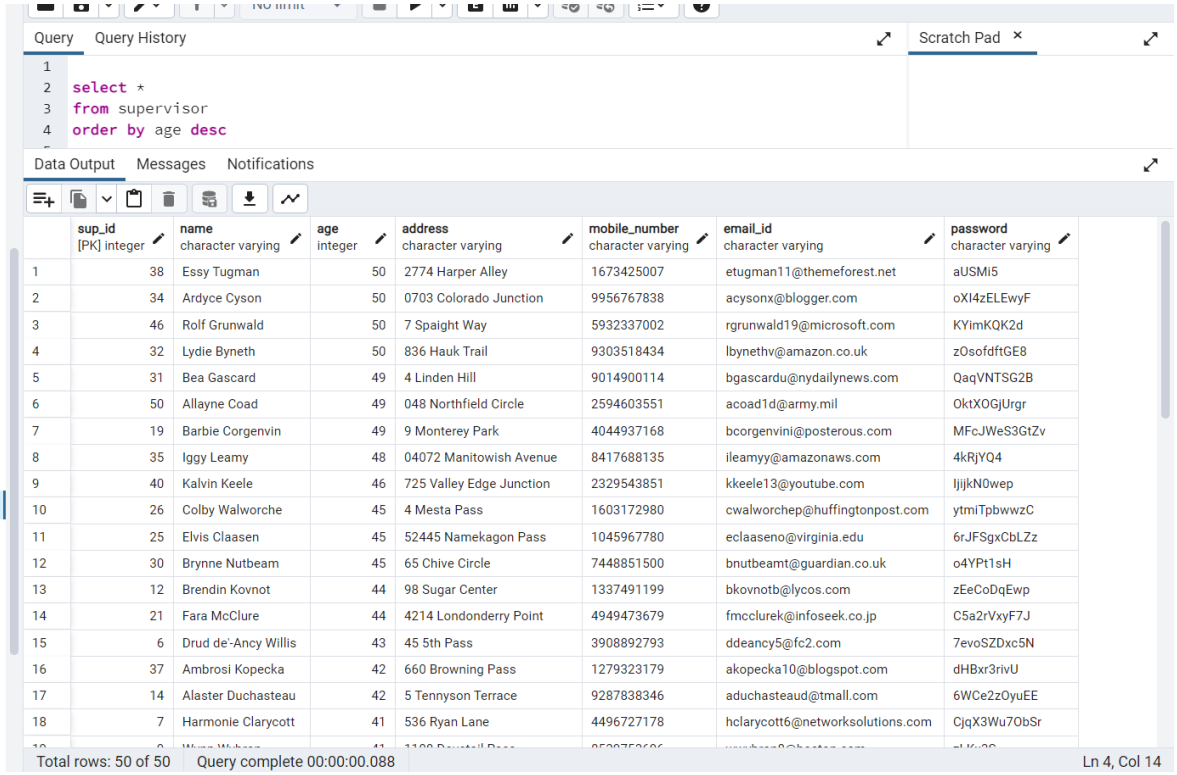
	count bigint
1	72

3. Give the data of the supervisor table in descending order of age.

   select *

   from supervisor

   order by age desc



Query | Query History

```
1
2  select *
3  from supervisor
4  order by age desc
```

Scratch Pad

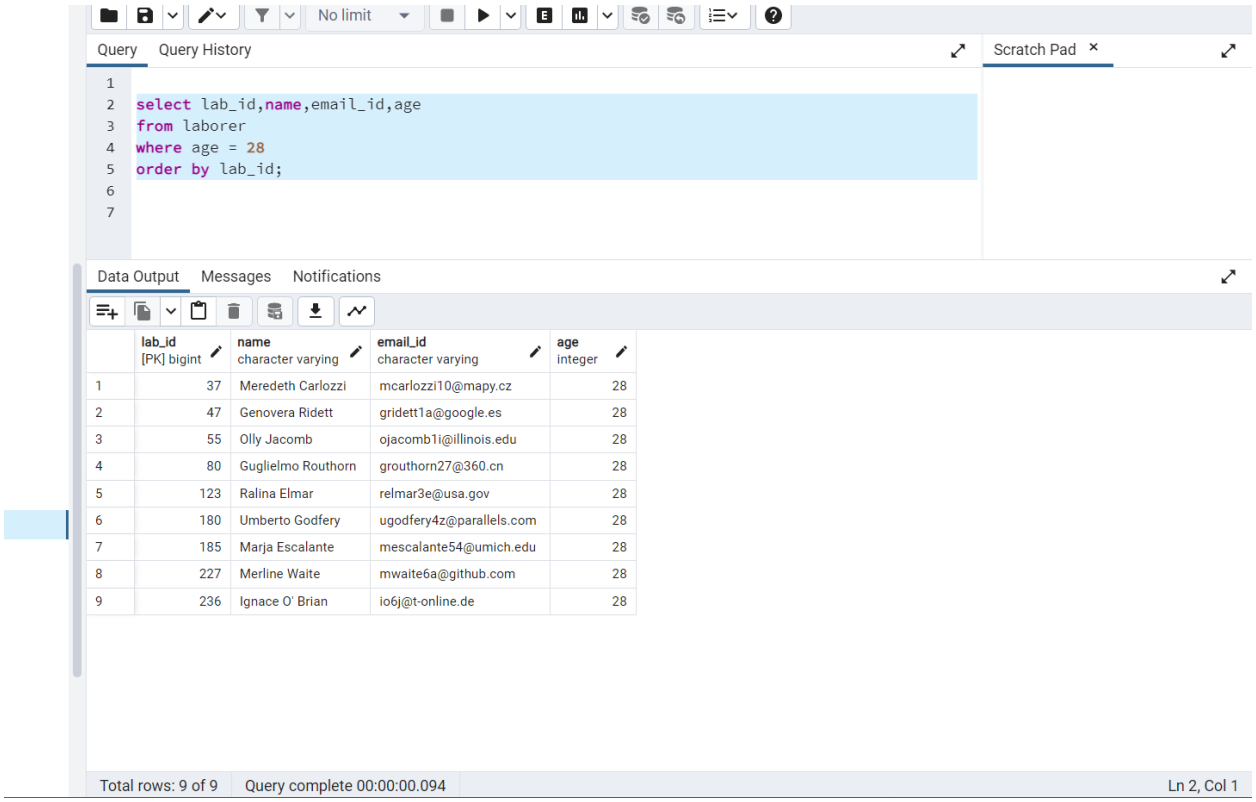
Data Output | Messages | Notifications

	sup_id [PK] integer	name character varying	age integer	address character varying	mobile_number character varying	email_id character varying	password character varying
1	38	Essy Tugman	50	2774 Harper Alley	1673425007	etugman11@themeforest.net	aUSMi5
2	34	Ardyce Cyson	50	0703 Colorado Junction	9956767838	acysonx@blogger.com	oXI4zELEwyF
3	46	Rolf Grunwald	50	7 Spaight Way	5932337002	rgrunwald19@microsoft.com	KYimKQK2d
4	32	Lydie Byneth	50	836 Hauk Trail	9303518434	lbynethv@amazon.co.uk	zOsofdftGE8
5	31	Bea Gascard	49	4 Linden Hill	9014900114	bgascardu@nydailynews.com	QaqVNTSG2B
6	50	Allayne Coad	49	048 Northfield Circle	2594603551	acoad1d@army.mil	OktXOGjUrgr
7	19	Barbie Corgenvin	49	9 Monterey Park	4044937168	bcorgenvini@posterous.com	MFcJWeS3GtZv
8	35	Iggy Leamy	48	04072 Manitowish Avenue	8417688135	ileamyy@amazonaws.com	4kRjYQ4
9	40	Kalvin Keele	46	725 Valley Edge Junction	2329543851	kkeele13@youtube.com	IjijkN0wep
10	26	Colby Walworche	45	4 Mesta Pass	1603172980	cwalworchep@huffingtonpost.com	ytmiTpbwwzC
11	25	Elvis Claasen	45	52445 Namekagon Pass	1045967780	eclaaseno@virginia.edu	6rJFSgxCbLZz
12	30	Brynne Nutbeam	45	65 Chive Circle	7448851500	bnutbeamt@guardian.co.uk	o4YPt1sH
13	12	Brendin Kovnot	44	98 Sugar Center	1337491199	bkovnotb@lycos.com	zEeCoDqEwp
14	21	Fara McClure	44	4214 Londonderry Point	4949473679	fmcclurek@infoseek.co.jp	C5a2rVxyF7J
15	6	Drud de'-Ancy Willis	43	45 5th Pass	3908892793	ddeancy5@fc2.com	7evoSZDxc5N
16	37	Ambrosi Kopecka	42	660 Browning Pass	1279323179	akopecka10@blogspot.com	dHBxr3rivU
17	14	Alaster Duchasteau	42	5 Tennyson Terrace	9287838346	aduchasteaud@tmall.com	6WCe2zOyuEE
18	7	Harmonie Clarycott	41	536 Ryan Lane	4496727178	hclarycott6@networksolutions.com	CjqX3Wu7ObSr

Total rows: 50 of 50 | Query complete 00:00:00.088 | Ln 4, Col 14

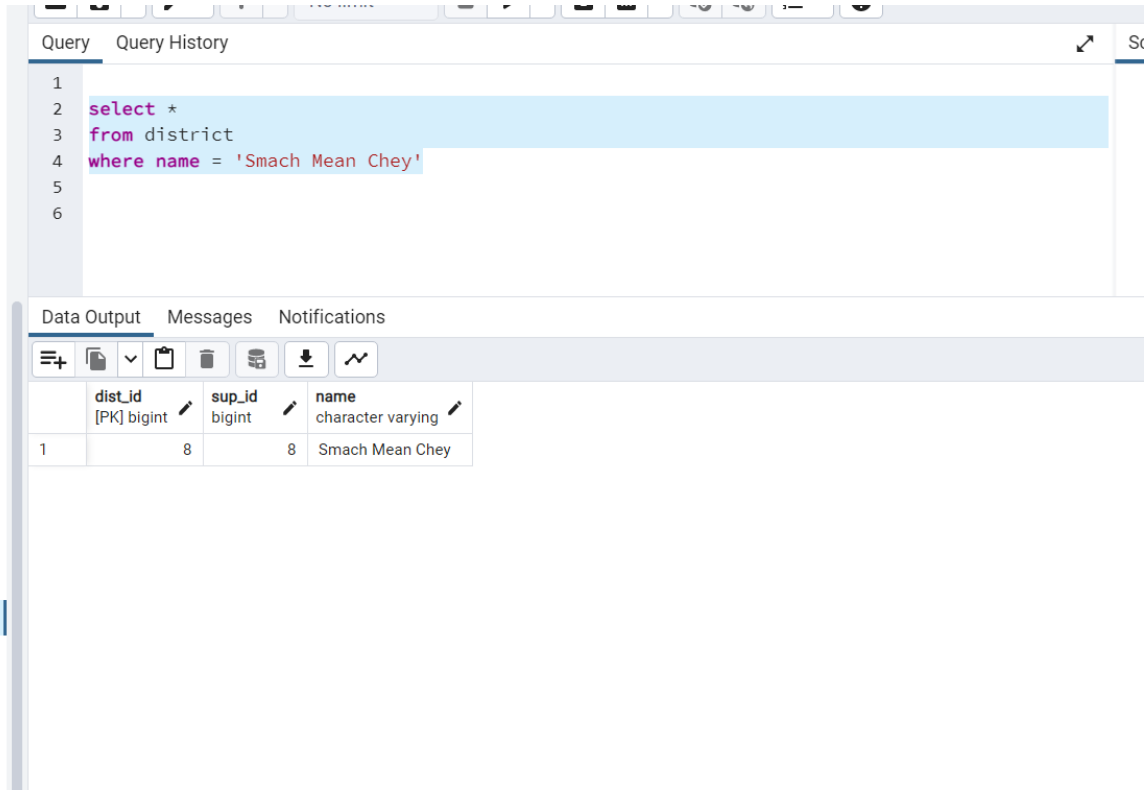
4. Find the name, id, and password of all laborers whose age is 28 in ascending order of their ids.

```
select lab_id,name,email_id,age
from laborer
where age = 28
order by lab_id;
```



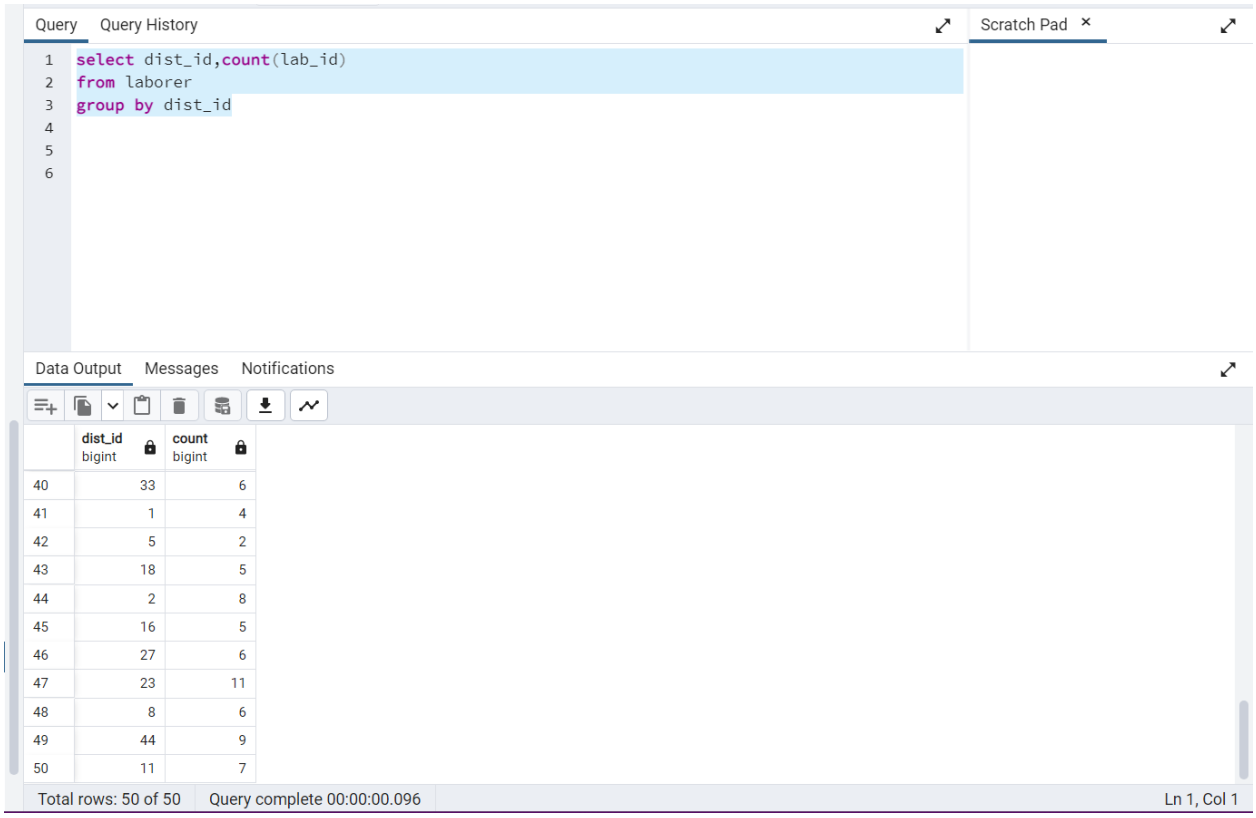
5. Find the data of the district whose name is Smach Mean Chey.

```
select *
from district
where name = 'Smach Mean Chey'
```



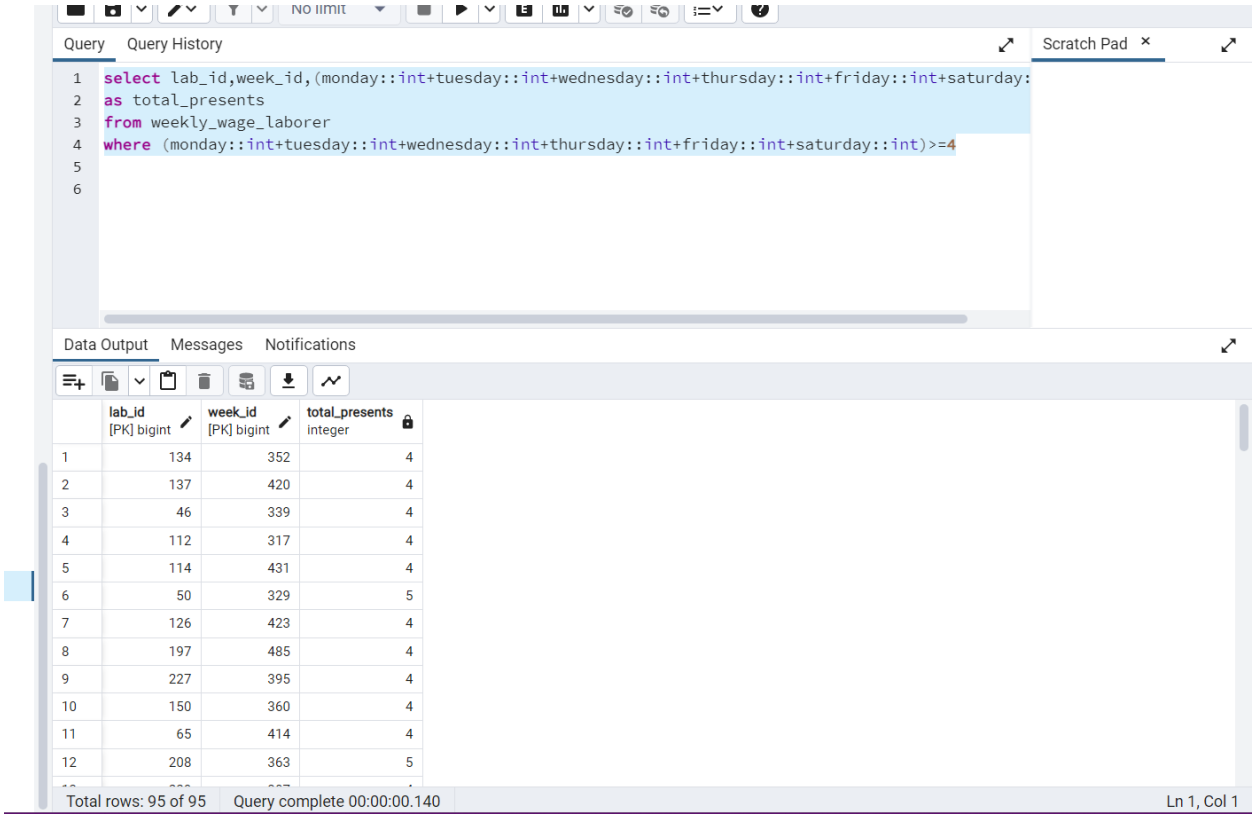
6. Find the number of laborers in the particular district.

select dist_id,count(lab_id)
from laborer
group by dist_id



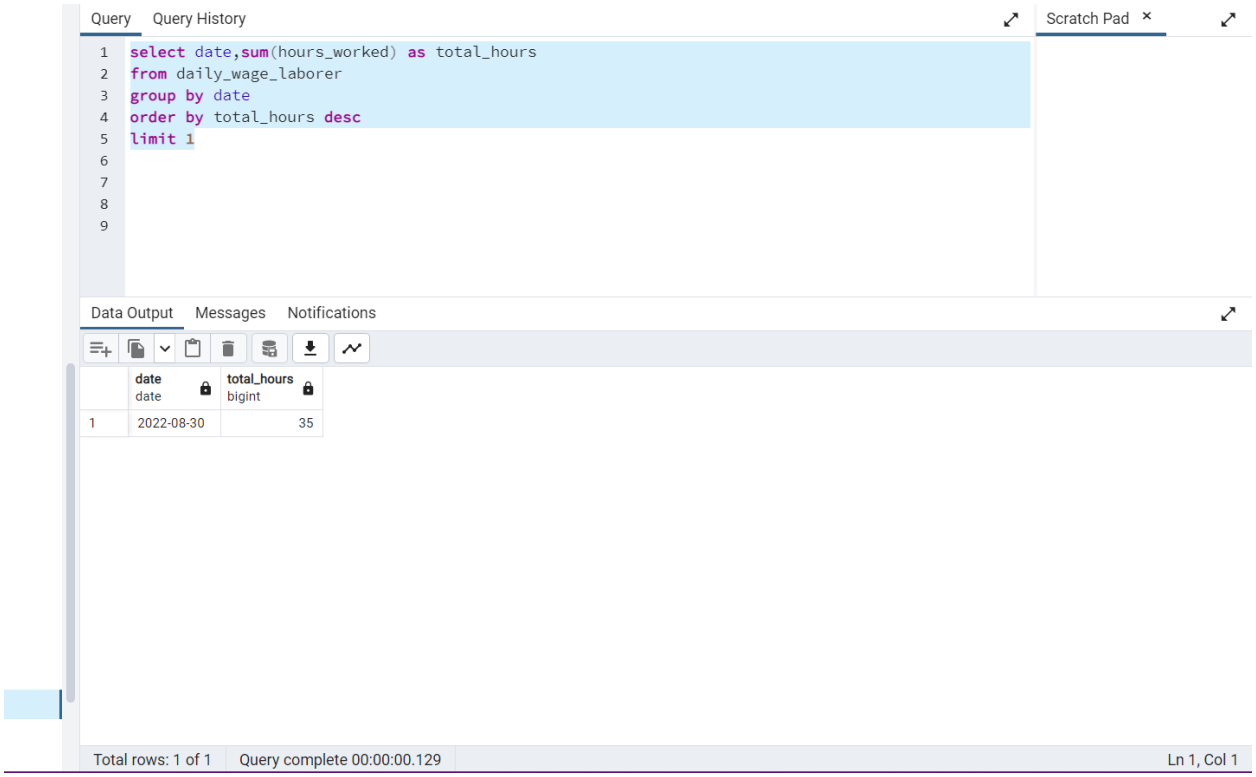
7. Filter out all the rows where present was recorded more than 4 days from the weekly wage laborer table.

```
select
lab_id,week_id,(monday::int+tuesday::int+wednesday::int+thursday::int+friday::int+saturday::int)
as total_presents
from weekly_wage_laborer
where
(monday::int+tuesday::int+wednesday::int+thursday::int+friday::int+saturday::int)>=4
```



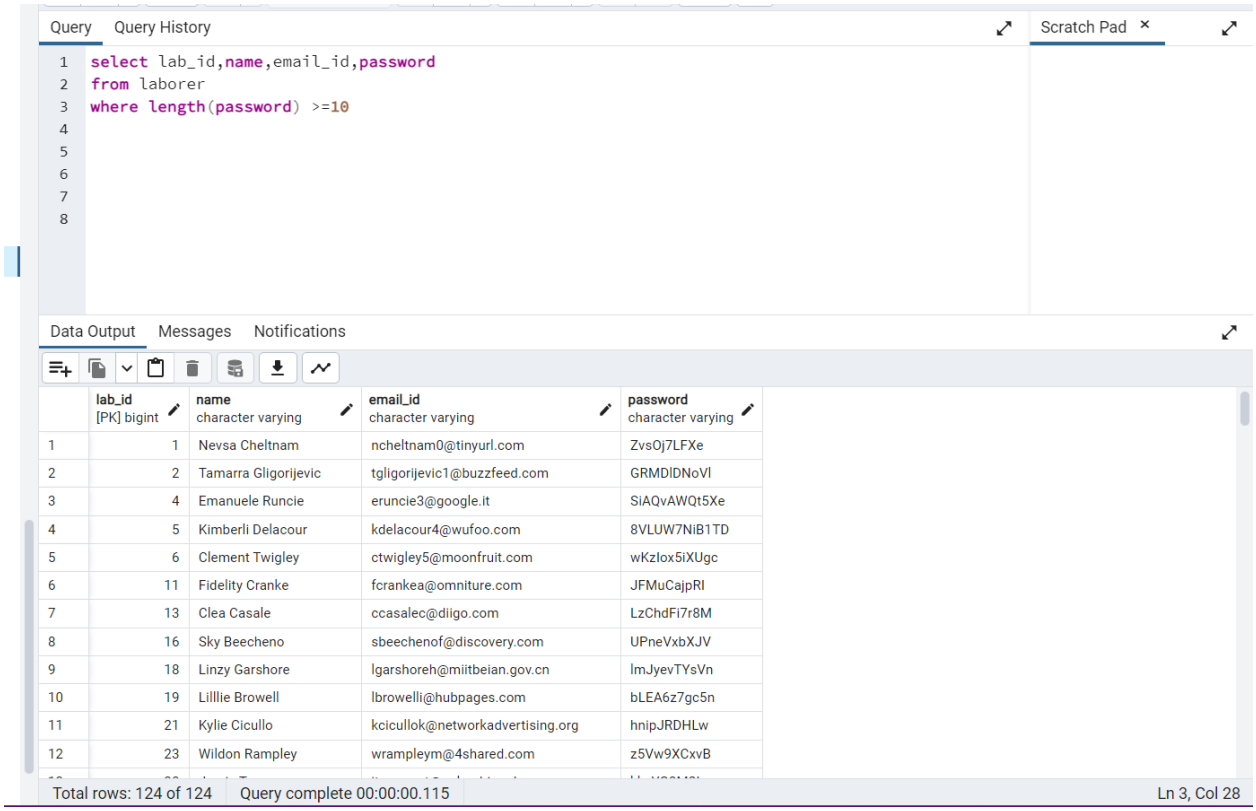
8. Find the date where maximum hours were given to all types of work

```
select date,sum(hours_worked) as total_hours
from daily_wage_laborer
group by date
order by total_hours desc
limit 1
```



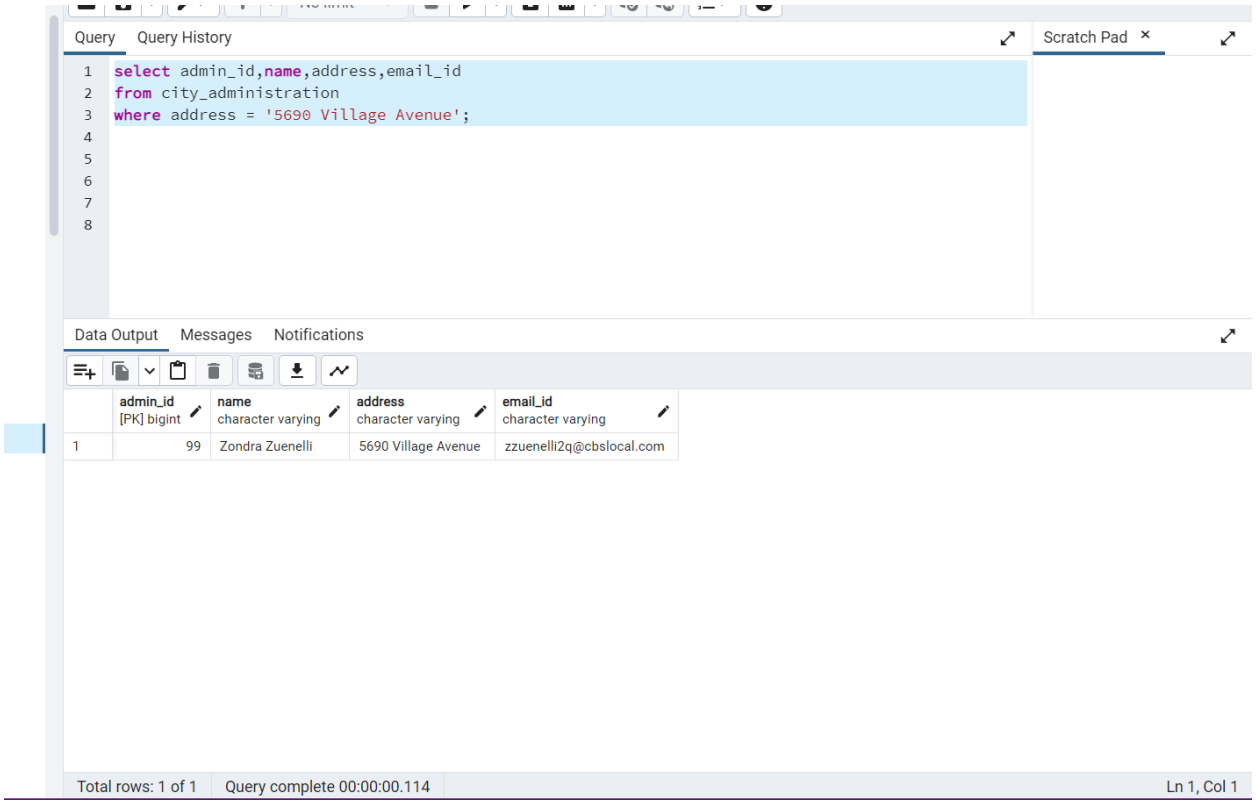
9. Find the laborers which have password length greater than or equal to 10.

```
select lab_id,name,email_id,password
from laborer
where length(password) >=10
```



10. List the city administrators whose address is “5690 Village Avenue”.

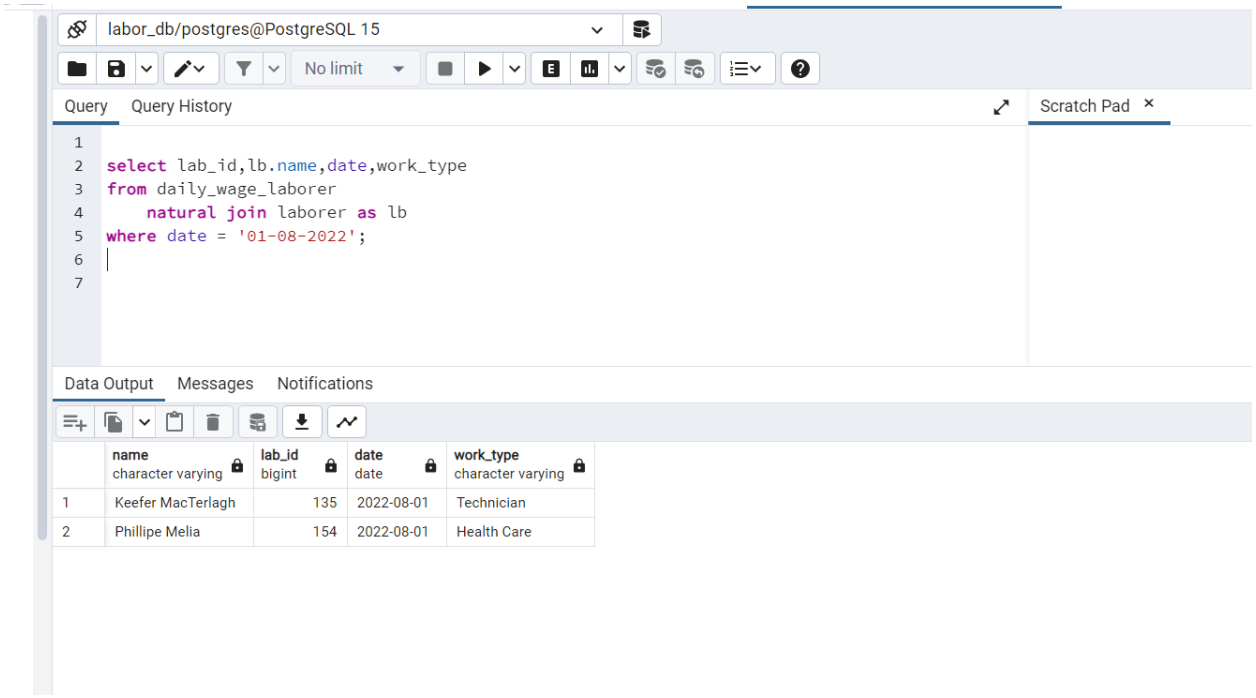
```
select admin_id, name, address,email_id
from city_administration
where address = '5690 Village Avenue';
```



3. Complex Queries

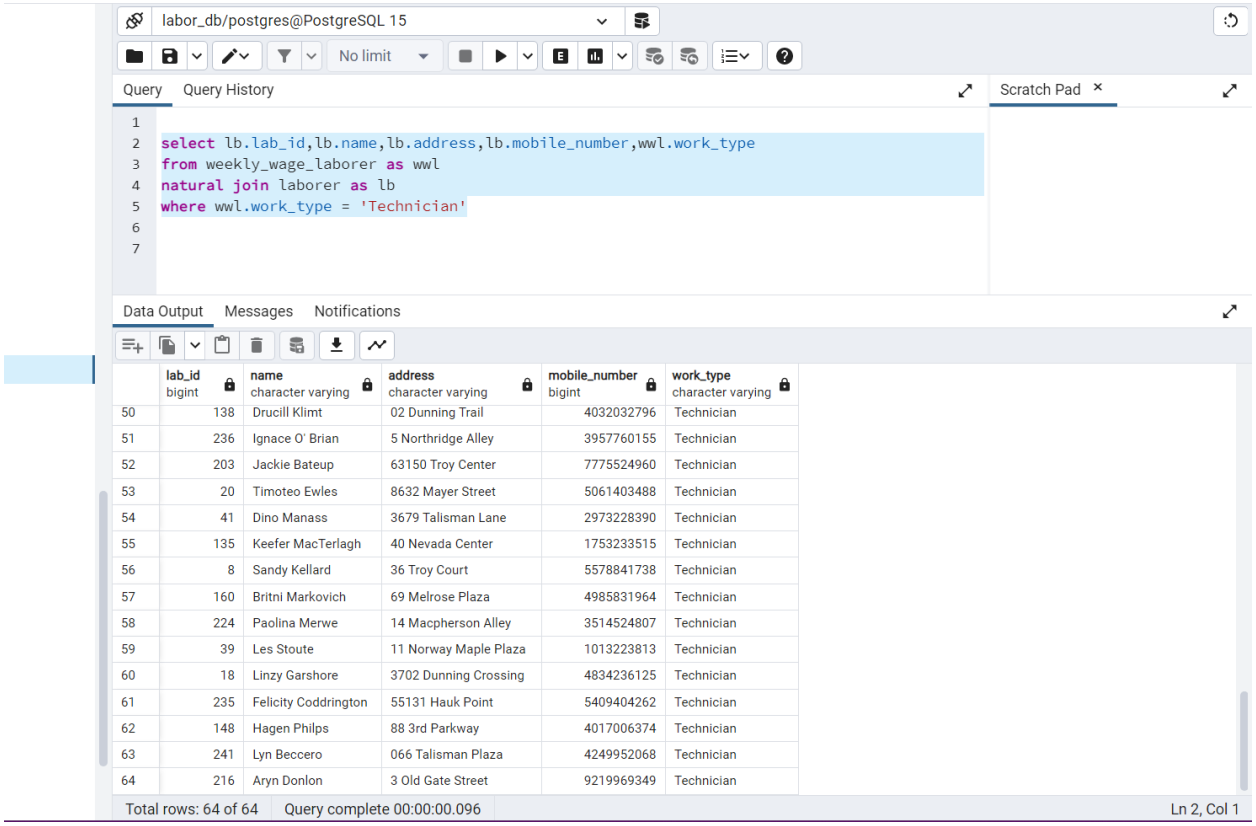
1. Find the lab id for which date is 01-08-2022 in daily wage laborer.

Select lab_id,lb_name,date,work_type
From daily_wage_laborer
Natural join laborer as lb
Where date = '01-08-2022';



2. Find all the laborers data who have worked as Technician weekly.

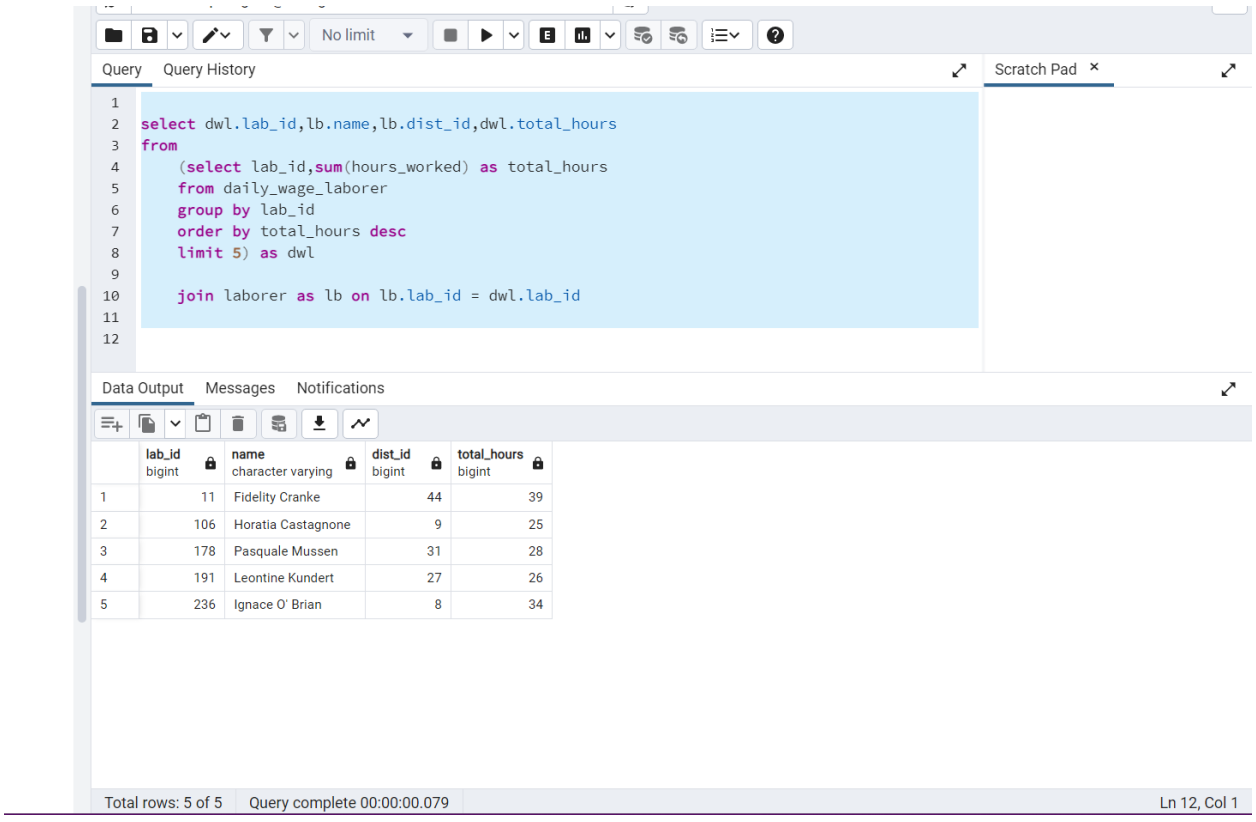
```
select lb.lab_id,lb.name,lb.address,lb.mobile_number,wwl.work_type
from weekly_wage_laborer as wwl
natural join laborer as lb
where wwl.work_type = 'Technician'
```



	lab_id bigint	name character varying	address character varying	mobile_number bigint	work_type character varying
50	138	Drucill Klimt	02 Dunning Trail	4032032796	Technician
51	236	Ignace O' Brian	5 Northridge Alley	3957760155	Technician
52	203	Jackie Bateup	63150 Troy Center	7775524960	Technician
53	20	Timoteo Ewles	8632 Mayer Street	5061403488	Technician
54	41	Dino Manass	3679 Talisman Lane	2973228390	Technician
55	135	Keefer MacTerlagh	40 Nevada Center	1753233515	Technician
56	8	Sandy Kellard	36 Troy Court	5578841738	Technician
57	160	Britni Markovich	69 Melrose Plaza	4985831964	Technician
58	224	Paolina Merwe	14 Macpherson Alley	3514524807	Technician
59	39	Les Stoute	11 Norway Maple Plaza	1013223813	Technician
60	18	Linzy Garshore	3702 Dunning Crossing	4834236125	Technician
61	235	Felicity Coddrington	55131 Hauk Point	5409404262	Technician
62	148	Hagen Philps	88 3rd Parkway	4017006374	Technician
63	241	Lyn Beccero	066 Talisman Plaza	4249952068	Technician
64	216	Aryn Donlon	3 Old Gate Street	9219969349	Technician

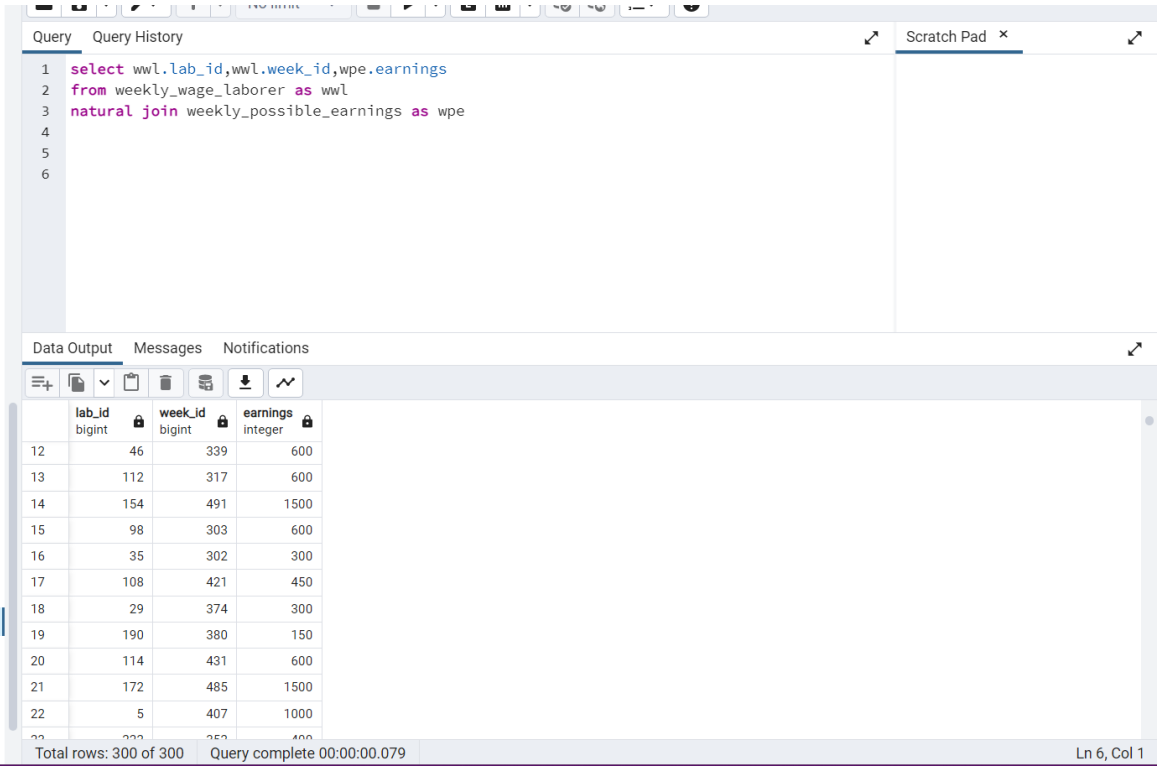
3. Find the name,dist_id and total_hours of work top 5 daily wage laborers who worked the most hourly.

```
select dwl.lab_id,lb.name,lb.dist_id,dwl.total_hours
from
        (select lab_id,sum(hours_worked) as total_hours
        from daily_wage_laborer
        group by lab_id
        order by total_hours desc
        limit 5) as dwl

join laborer as lb on lb.lab_id = dwl.lab_id
```



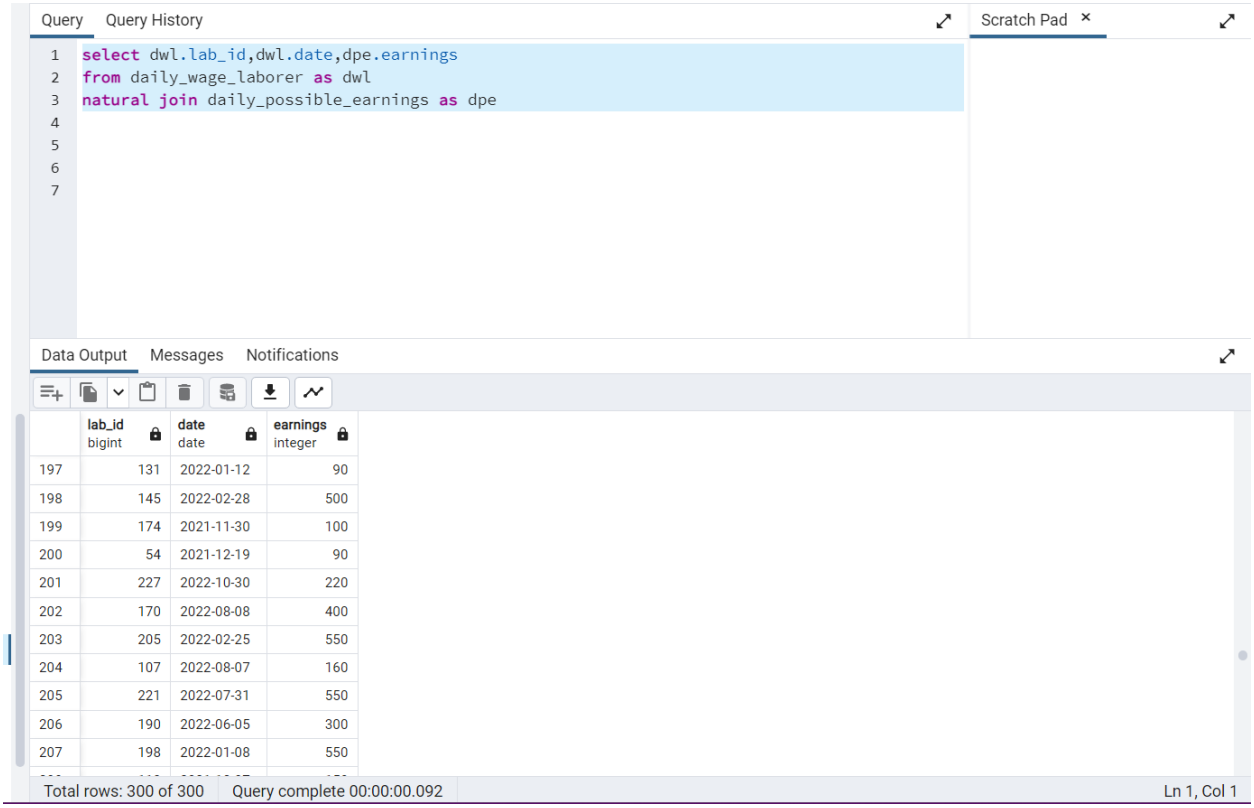
4. Find the earnings of the laborer in a particular week.

select wwl.lab_id,wwl.week_id,wpe.earnings
from weekly_wage_laborer as wwl
natural join weekly_possible_earnings as wpe



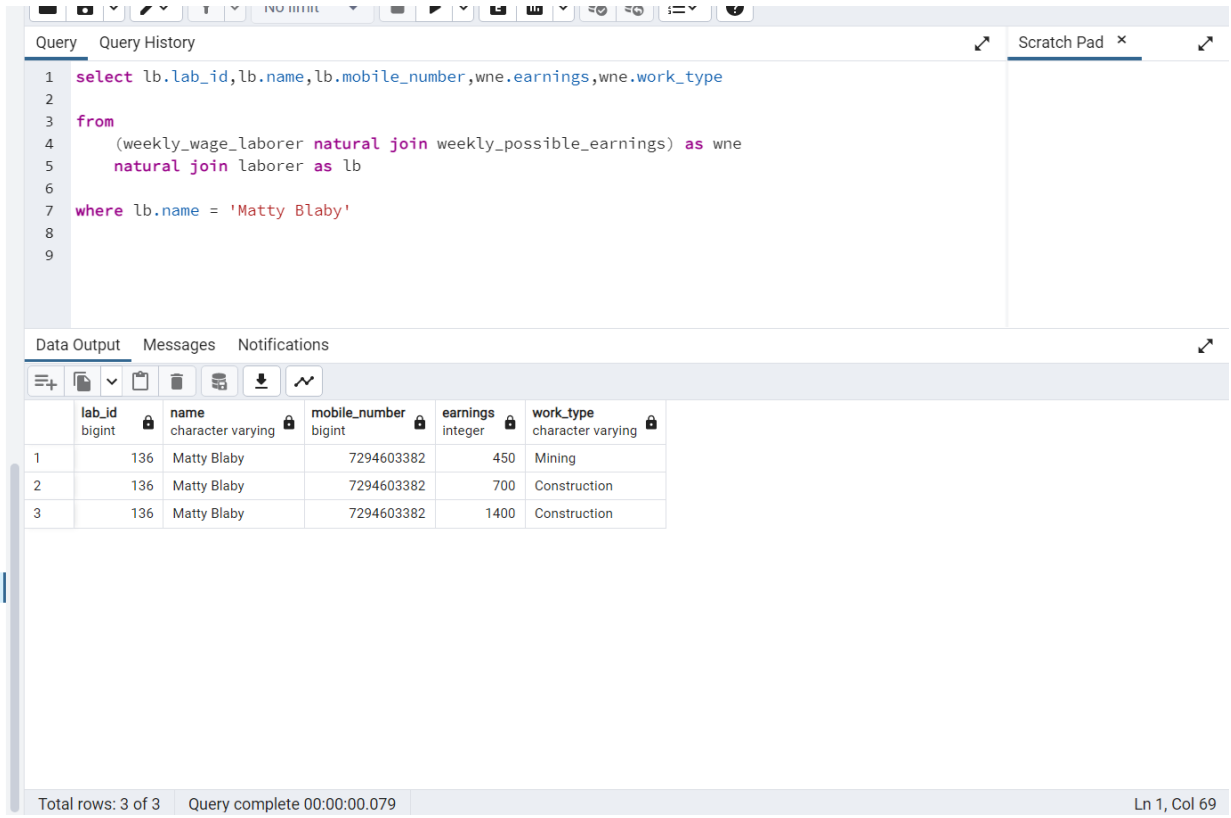
5. Find the earnings of the laborer on a particular date.

```
select dwl.lab_id,dwl.date,dpe.earnings
from daily_wage_laborer as dwl
natural join daily_possible_earnings as dpe
```



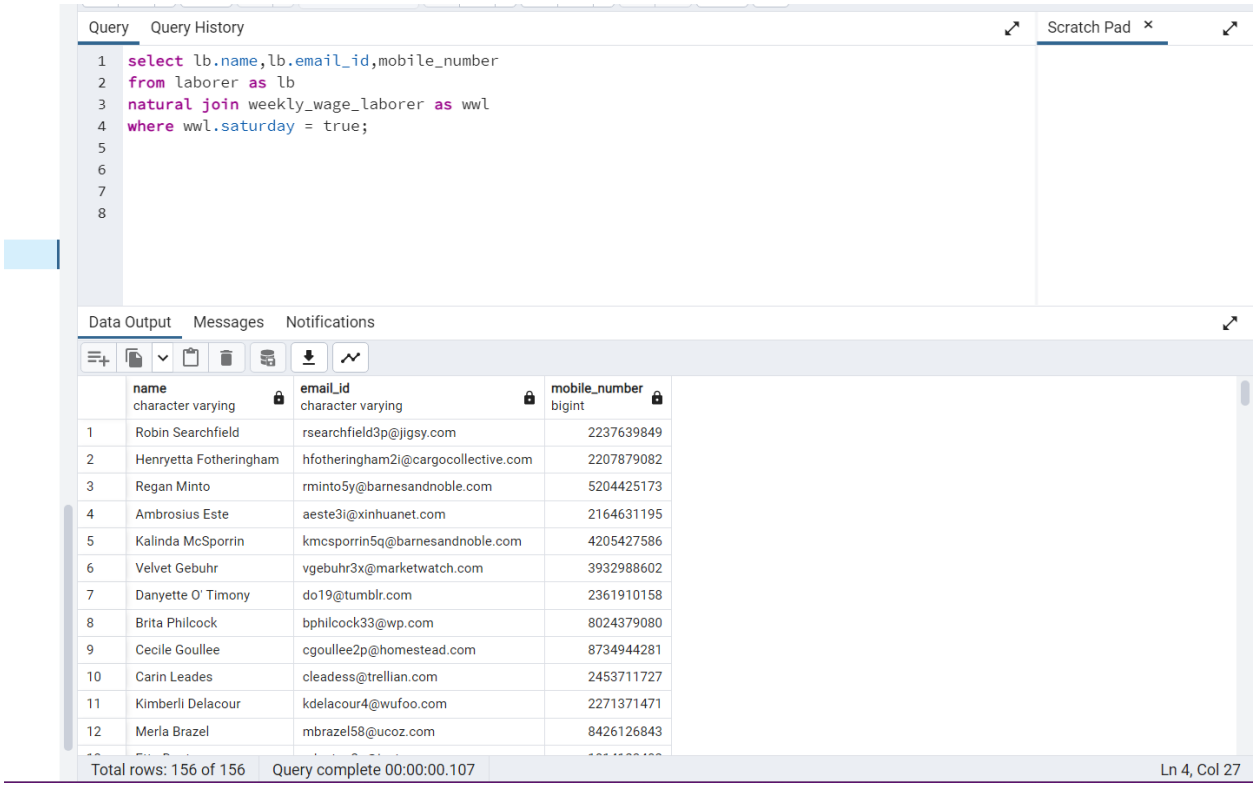
6. Find all the weekly earnings (if there) of laborer “Matty Blaby”.

```
select lb.lab_id,lb.name,lb.mobile_number,wne.earnings,wne.work_type

from
        (weekly_wage_laborer natural join weekly_possible_earnings) as wne
        natural join laborer as lb

where lb.name = 'Matty Blaby'
```



7. Find the data of laborers which were present on any saturday.

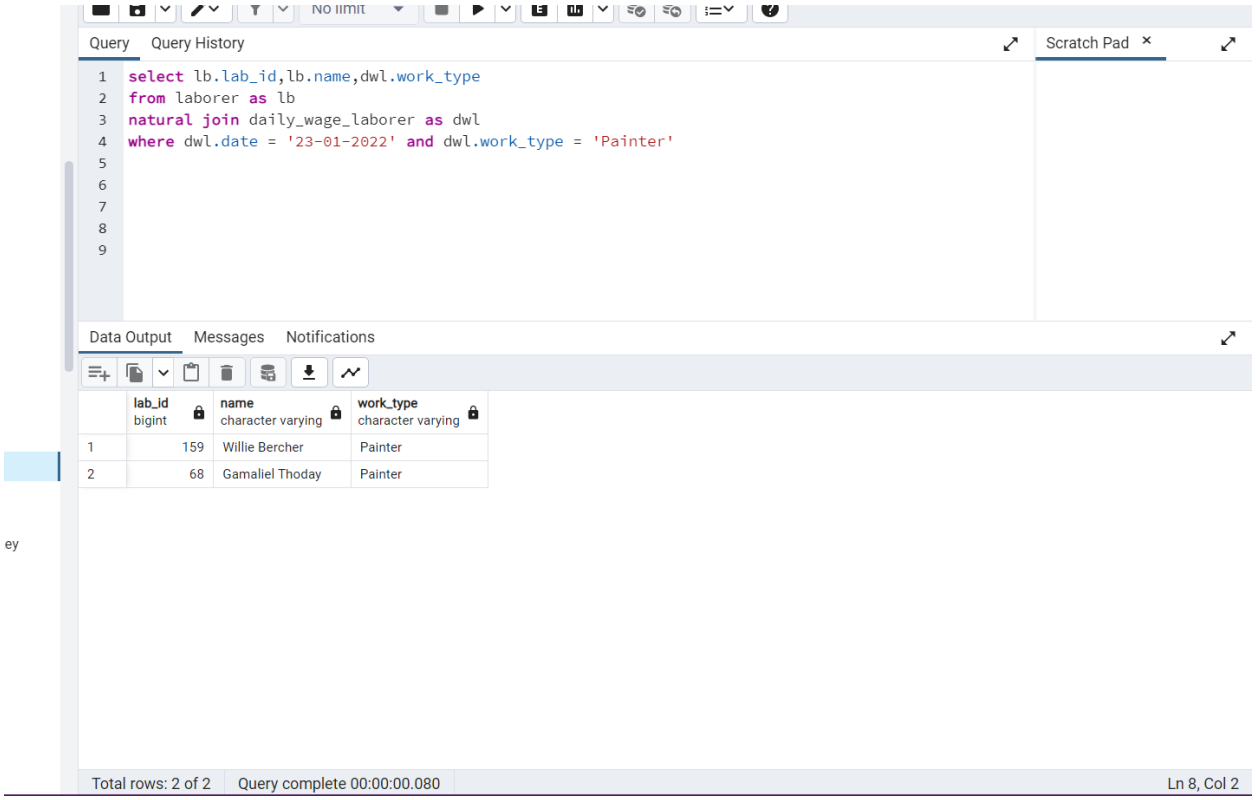
```
select lb.name,lb.email_id,mobile_number
from laborer as lb
natural join weekly_wage_laborer as wwl
where wwl.saturday = true;
```



	name character varying	email_id character varying	mobile_number bigint
1	Robin Searchfield	rsearchfield3p@jigsy.com	2237639849
2	Henryetta Fotheringham	hfotheringham2i@cargocollective.com	2207879082
3	Regan Minto	rminto5y@barnesandnoble.com	5204425173
4	Ambrosius Este	aeste3i@xinhuanet.com	2164631195
5	Kalinda McSporrin	kmcsporrin5q@barnesandnoble.com	4205427586
6	Velvet Gebuhr	vgebuhr3x@marketwatch.com	3932988602
7	Danyette O' Timony	do19@tumblr.com	2361910158
8	Brita Philcock	bphilcock33@wp.com	8024379080
9	Cecile Goullee	cgoullee2p@homestead.com	8734944281
10	Carin Leades	cleadess@trellian.com	2453711727
11	Kimberli Delacour	kdelacour4@wufoo.com	2271371471
12	Merla Brazel	mbrazel58@ucoz.com	8426126843

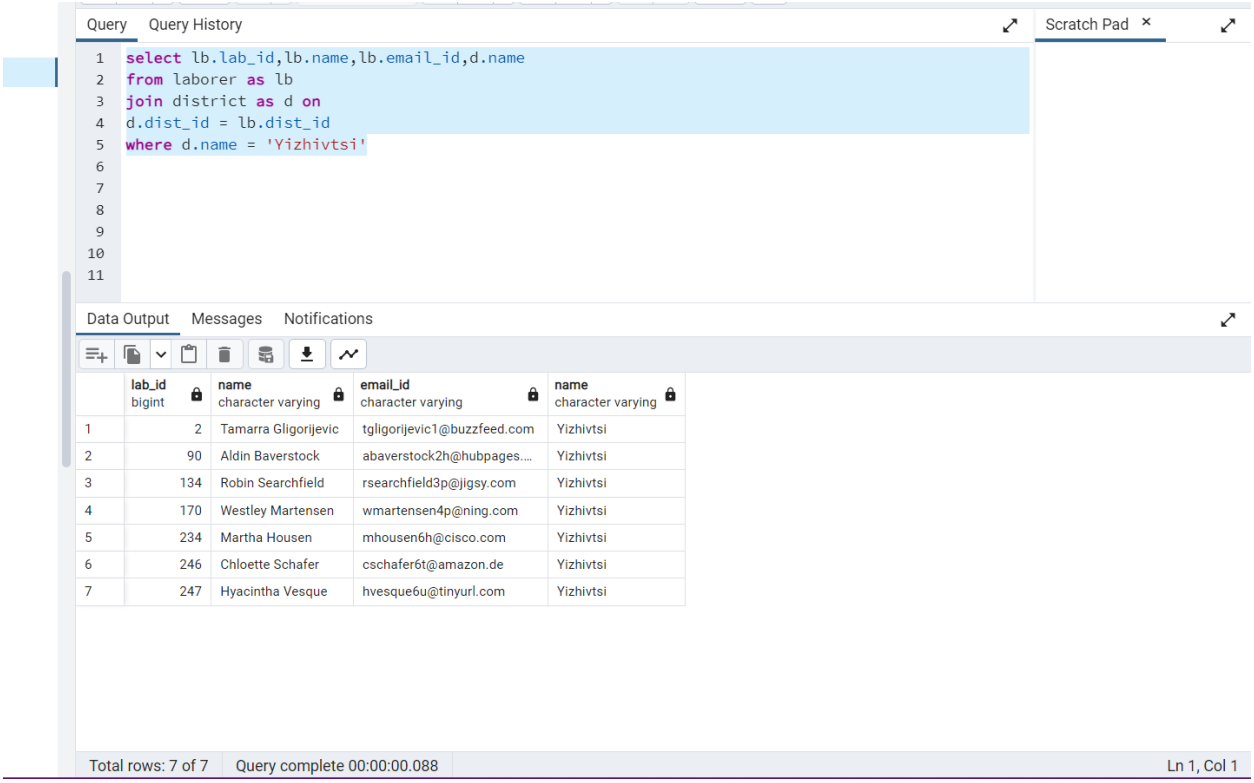
8. Find the information of laborers who did painting work on ‘23-01-2022’’.

```
select lb.lab_id,lb.name,dwl.work_type
from laborer as lb
natural join daily_wage_laborer as dwl
where dwl.date = '23-01-2022' and dwl.work_type = 'Painter'
```



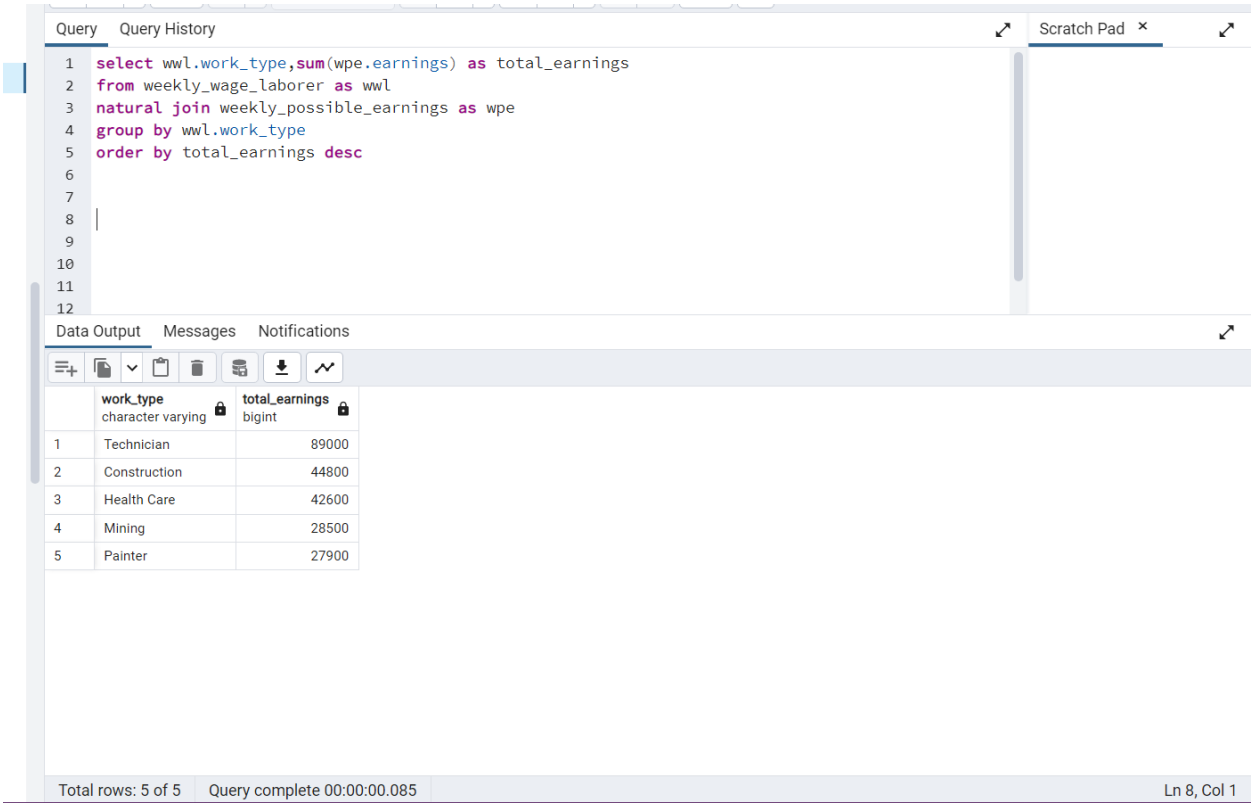
9. Find the information of laborers working in the district 'Yizhivtsi'

```
select lb.lab_id,lb. name,lb.email_id,d.name
from laborer as lb
join district as d on
d.dist_id = lb.dist_id
where d.name = 'Yizhivtsi'
```



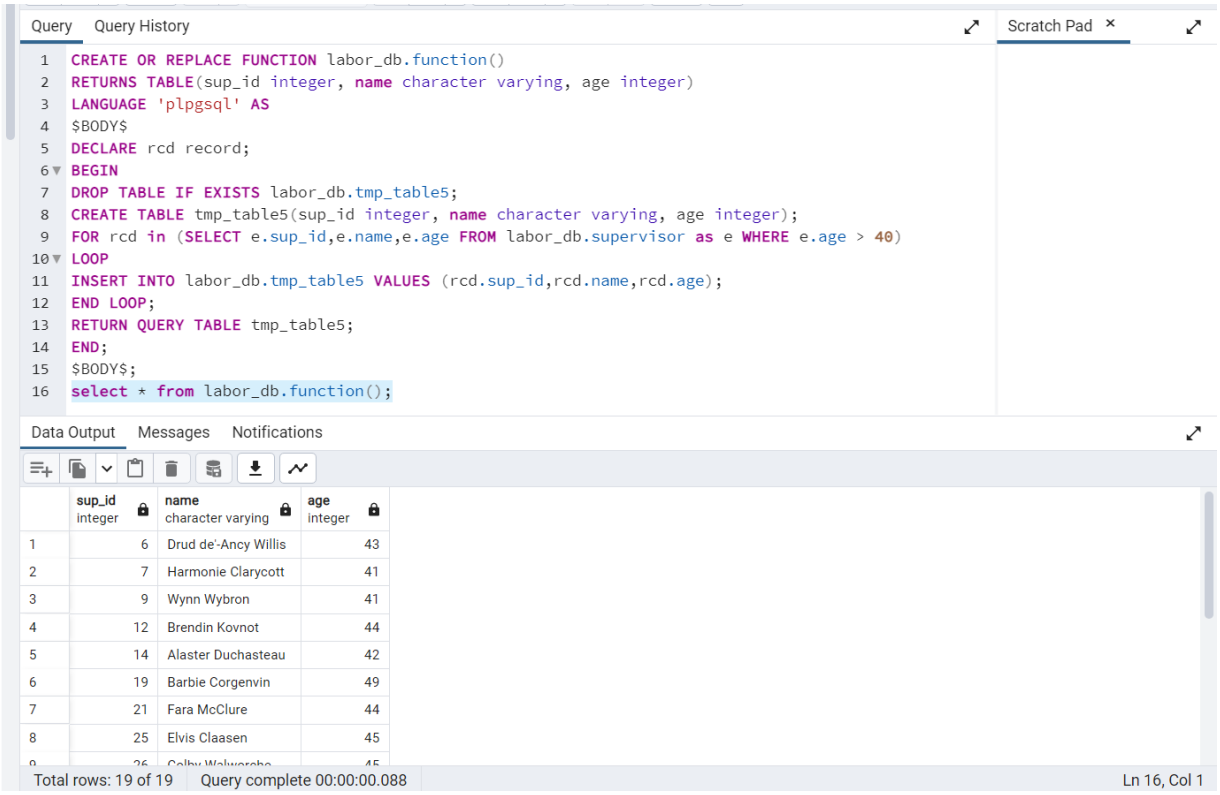
10. Sort the professions in descending order on the basis of their total earnings over all weeks.

```
select wwl.work_type,sum(wpe.earnings) as total_earnings
from weekly_wage_laborer as wwl
natural join weekly_possible_earnings as wpe
group by wwl.work_type
order by total_earnings desc
```



4. Function:

Function returns table with sup_id, name, age of supervisors which have age greater than 40.

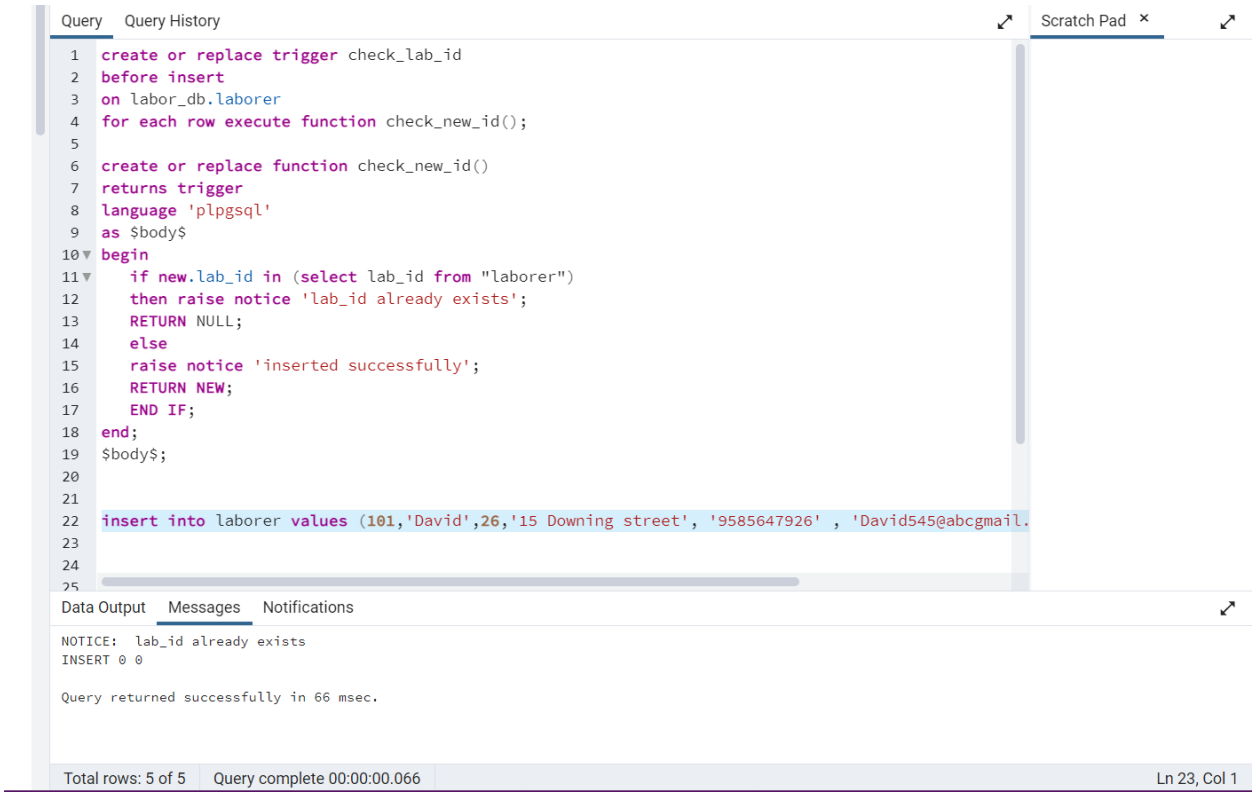



```
CREATE OR REPLACE FUNCTION labor_db.function()
RETURNS TABLE(sup_id integer, name character varying, age integer)
LANGUAGE 'plpgsql' AS
$BODY$
DECLARE rcd record;
BEGIN
DROP TABLE IF EXISTS labor_db.tmp_table5;
CREATE TABLE tmp_table5(sup_id integer, name character varying, age integer);
FOR rcd in (SELECT e.sup_id,e.name,e.age FROM labor_db.supervisor as e WHERE e.age > 40)
LOOP
INSERT INTO labor_db.tmp_table5 VALUES (rcd.sup_id,rcd.name,rcd.age);
```

```
END LOOP;
RETURN QUERY TABLE tmp_table5;
END;
$BODY$;
select * from labor_db.function();
```

5. Trigger:

Trigger on laborer to check if the Primary Key ID already exists or not before inserting a new record & send a custom reply instead of an error message.




```
create or replace trigger check_lab_id
before insert
on labor_db.laborer
for each row execute function check_new_id();
```

```
create or replace function check_new_id()
returns trigger
language 'plpgsql'
as $body$
begin
  if new.lab_id in (select lab_id from "laborer")
  then raise notice 'lab_id already exists';
  RETURN NULL;
  else
  raise notice 'inserted successfully';
  RETURN NEW;
  END IF;
end;
$body$;


insert into laborer values (101,'David',26,'15 Downing street', '9585647926' ,
'David545@abcgmail.com', 'DavidJack545', 25);
```

V. Website Designing

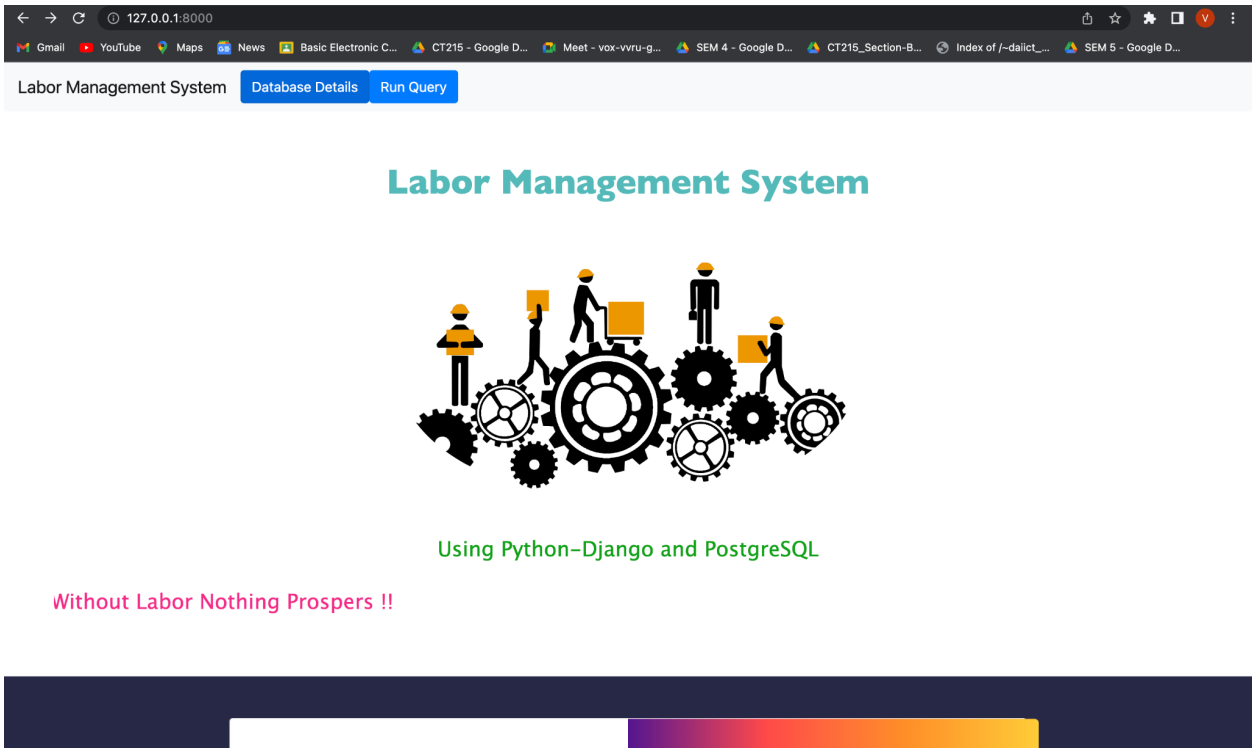
Laborer management website (Front-End + Back-end)

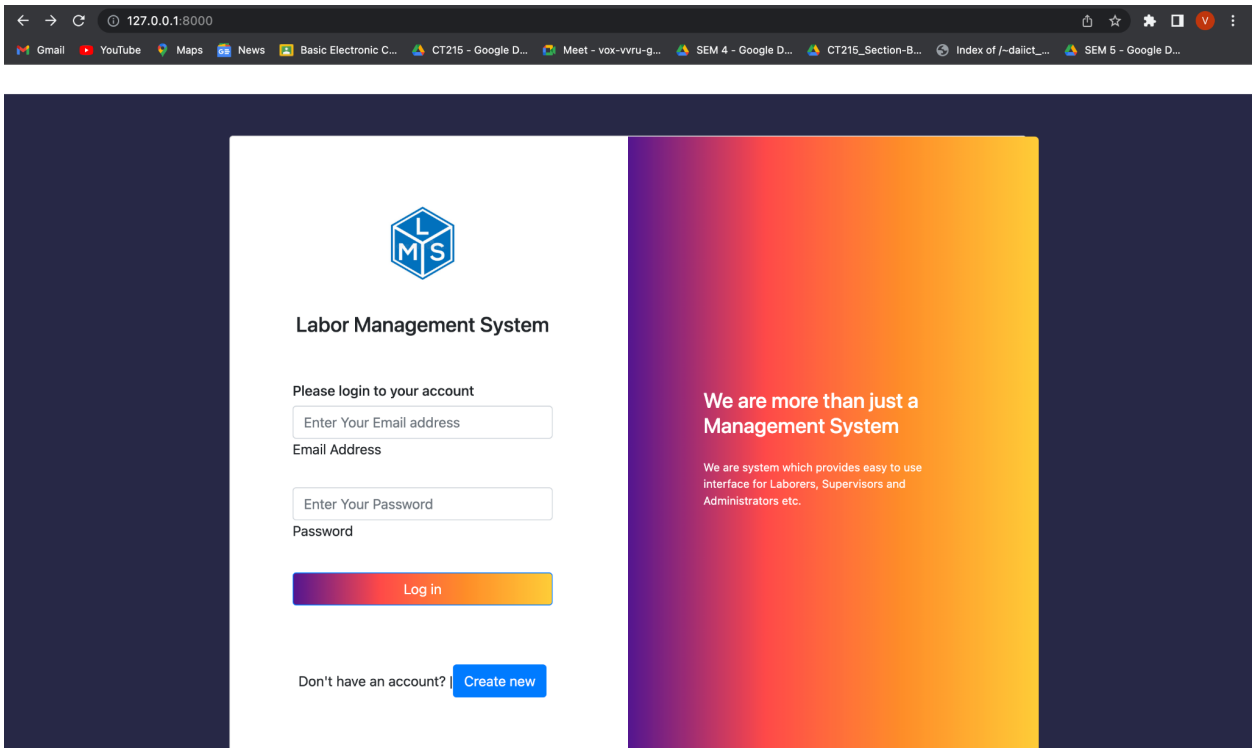
⇒We built our laborer management system with certain functionalities.

1. Sorting of tables with respect to attributes
2. Updation of Table records
3. Deletion of Table records
4. Insertion of valid records

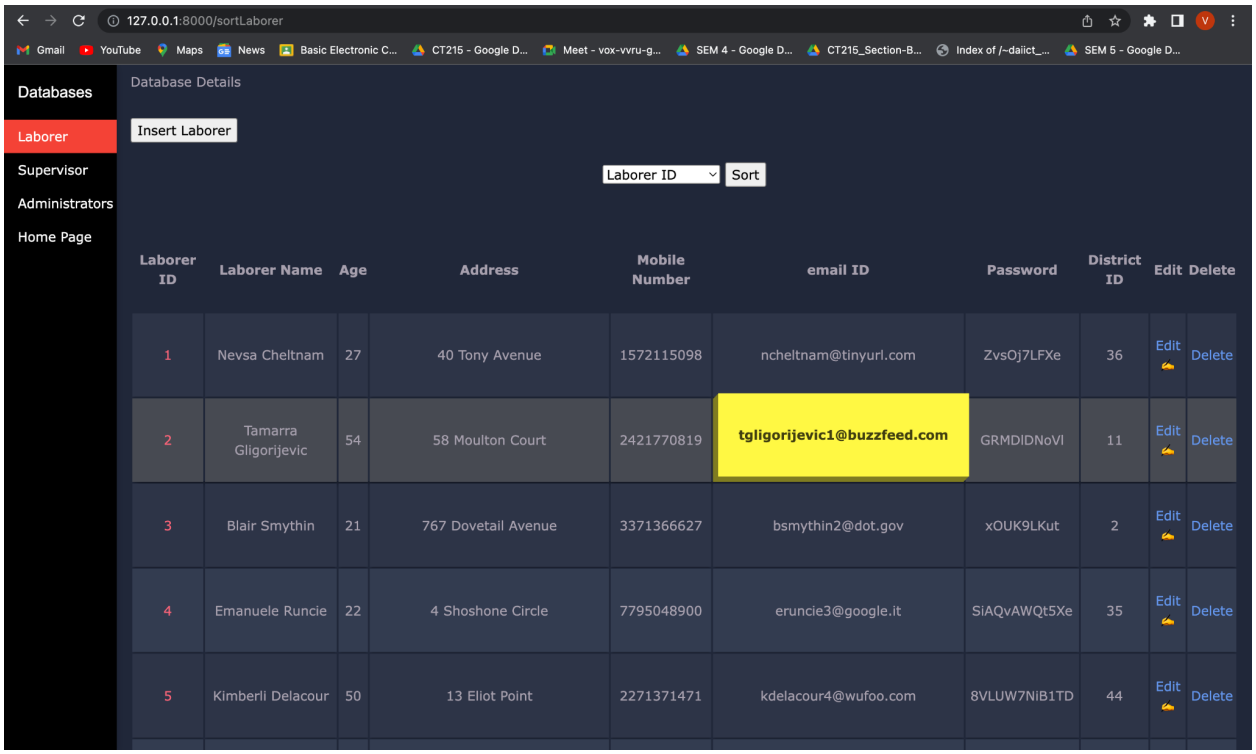
GitHub Repository Link: https://github.com/vanshp2002/Labor-Management-App.git

Home Page

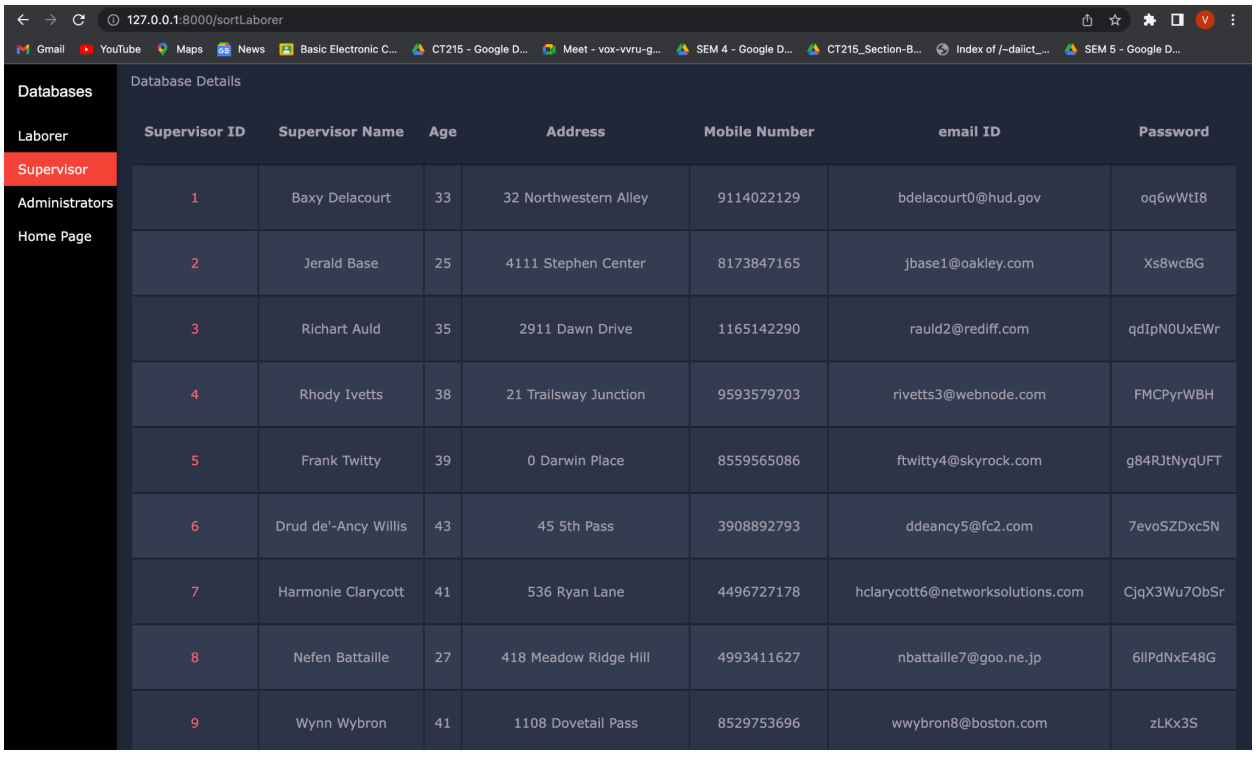

127.0.0.1:8000
Gmail
YouTube
Maps
News
Basic Electronic C...
CT215 - Google D...
Meet - vox-vvru-g...
SEM 4 - Google D...
CT215_Section-B...
Index of /~daiict_...
SEM 5 - Google D...
Labor Management System
Database Details
Run Query
Labor Management System
Using Python-Django and PostgreSQL
Without Labor Nothing Prospers !!


127.0.0.1:8000
Gmail
YouTube
Maps
News
Basic Electronic C...
CT215 - Google D...
Meet - vox-vvru-g...
SEM 4 - Google D...
CT215_Section-B...
Index of /~daiict_...
SEM 5 - Google D...
Labor Management System
Please login to your account
Enter Your Email address
Email Address
Enter Your Password
Password
Log in
Don't have an account?
Create new
We are more than just a Management System
We are system which provides easy to use interface for Laborers, Supervisors and Administrators etc.

Laborer Database page

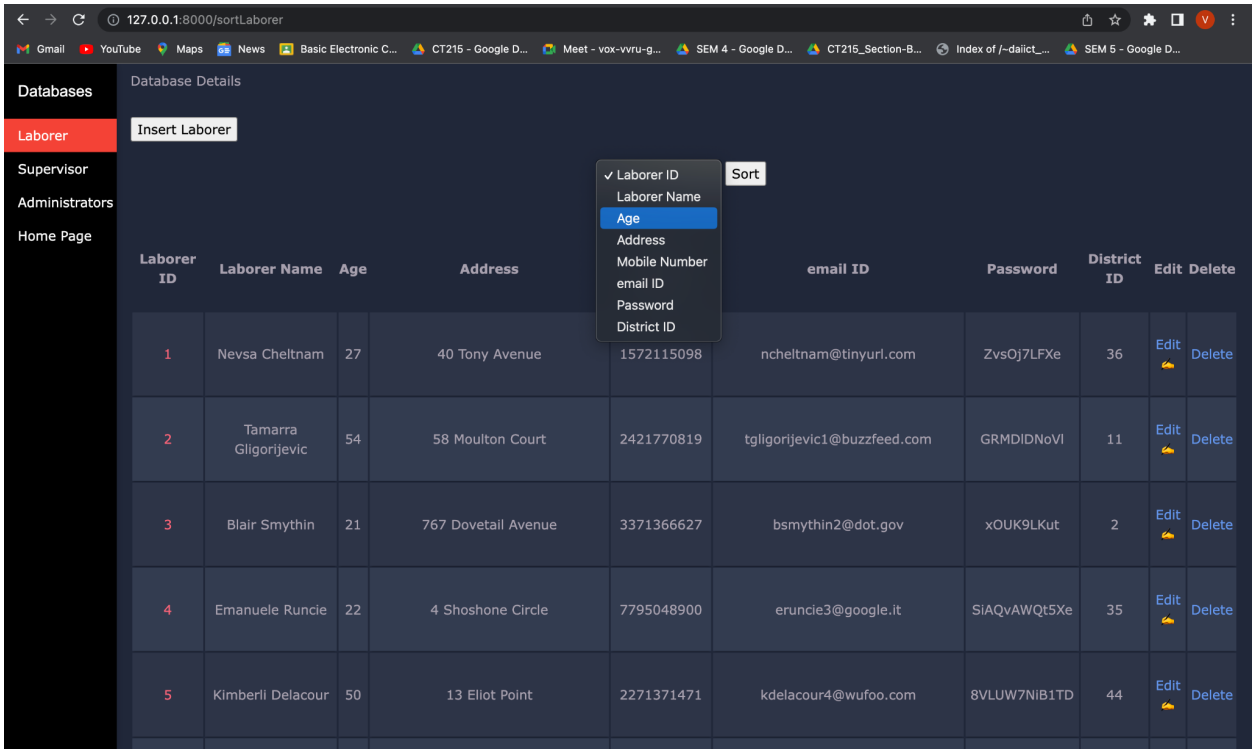


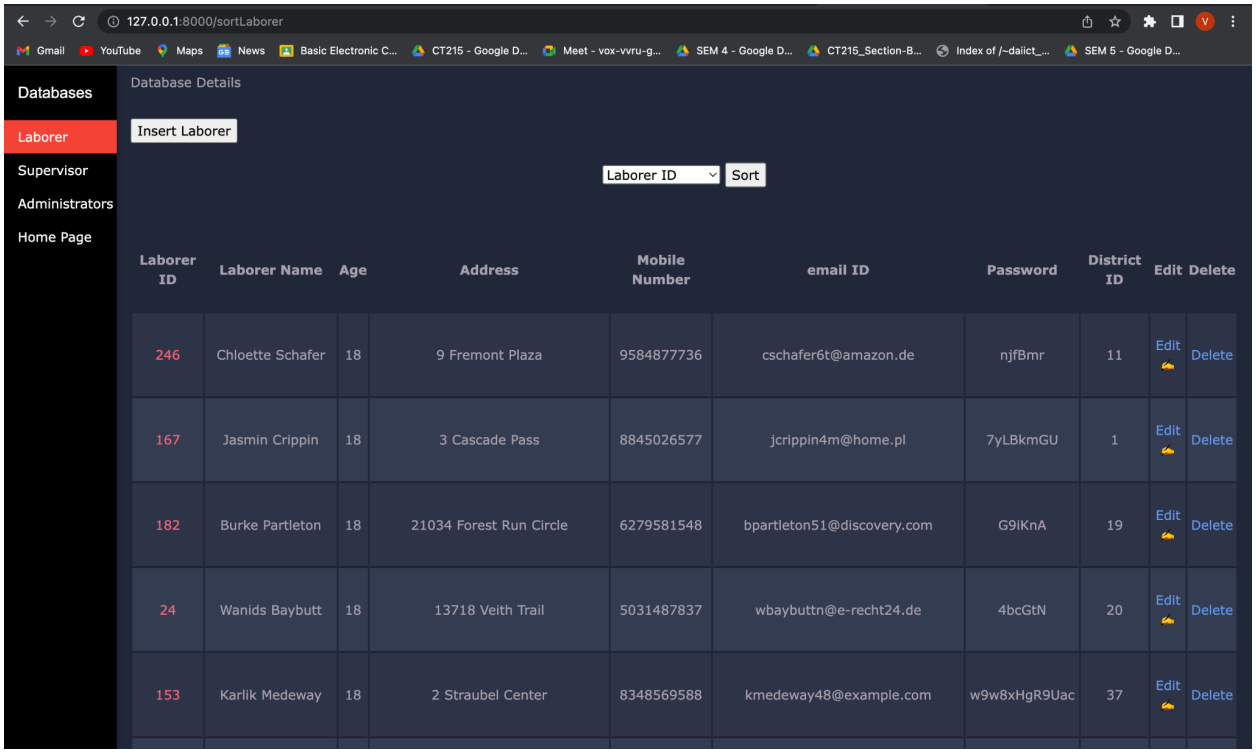
Supervisor Database page



Sorting the Laborer table on the basis of Age

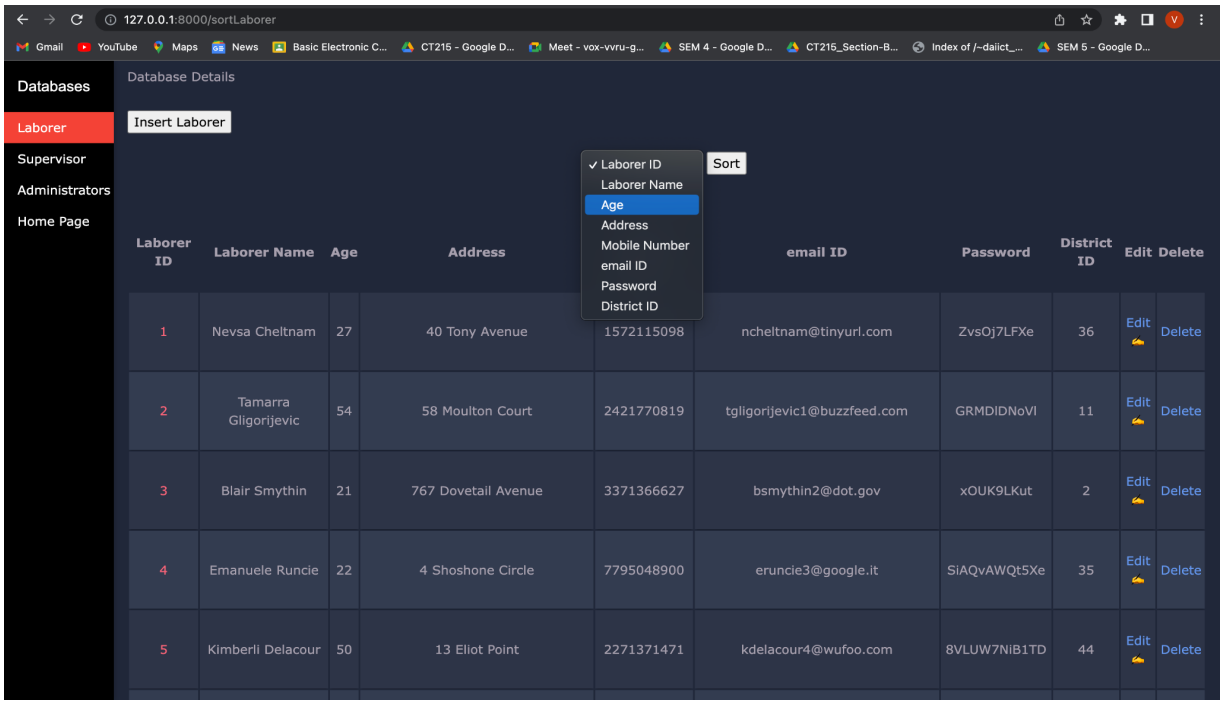
⇒ Here we sort our laborer table with respect to the ages of laborers and this table changes directly reflect the web page table and Postgres database.



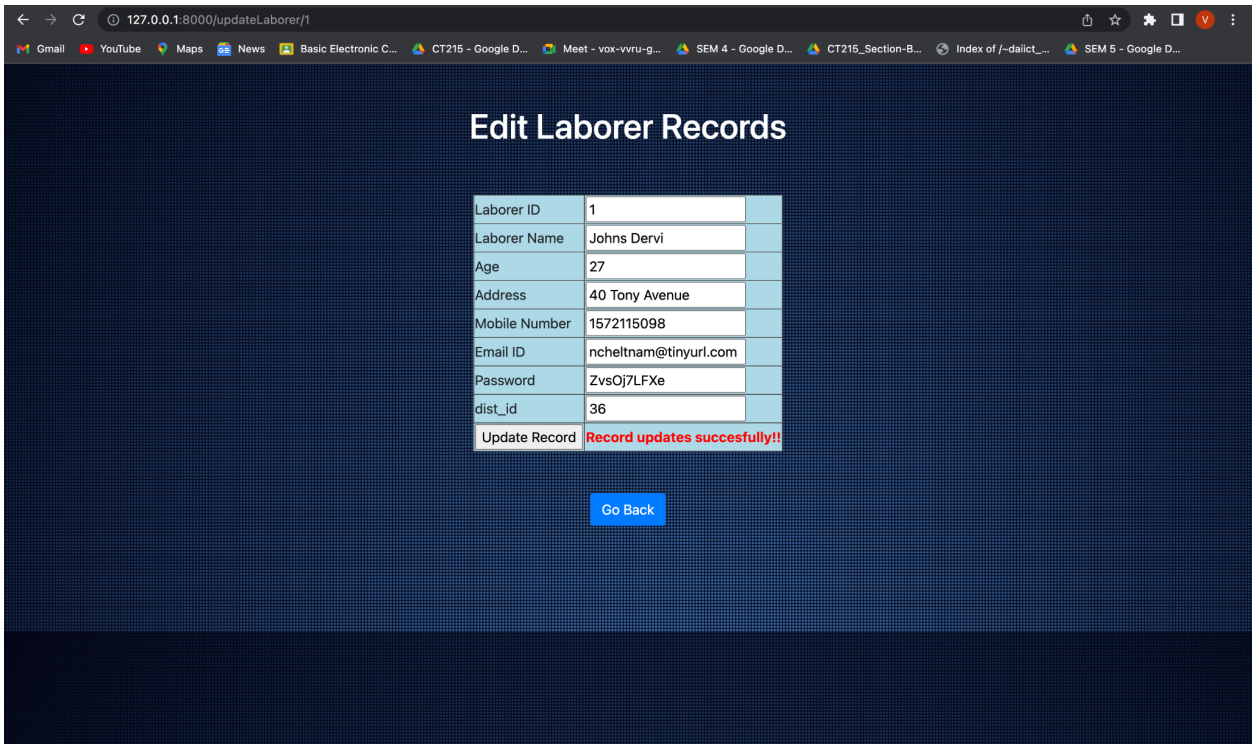


Edit operation on tables

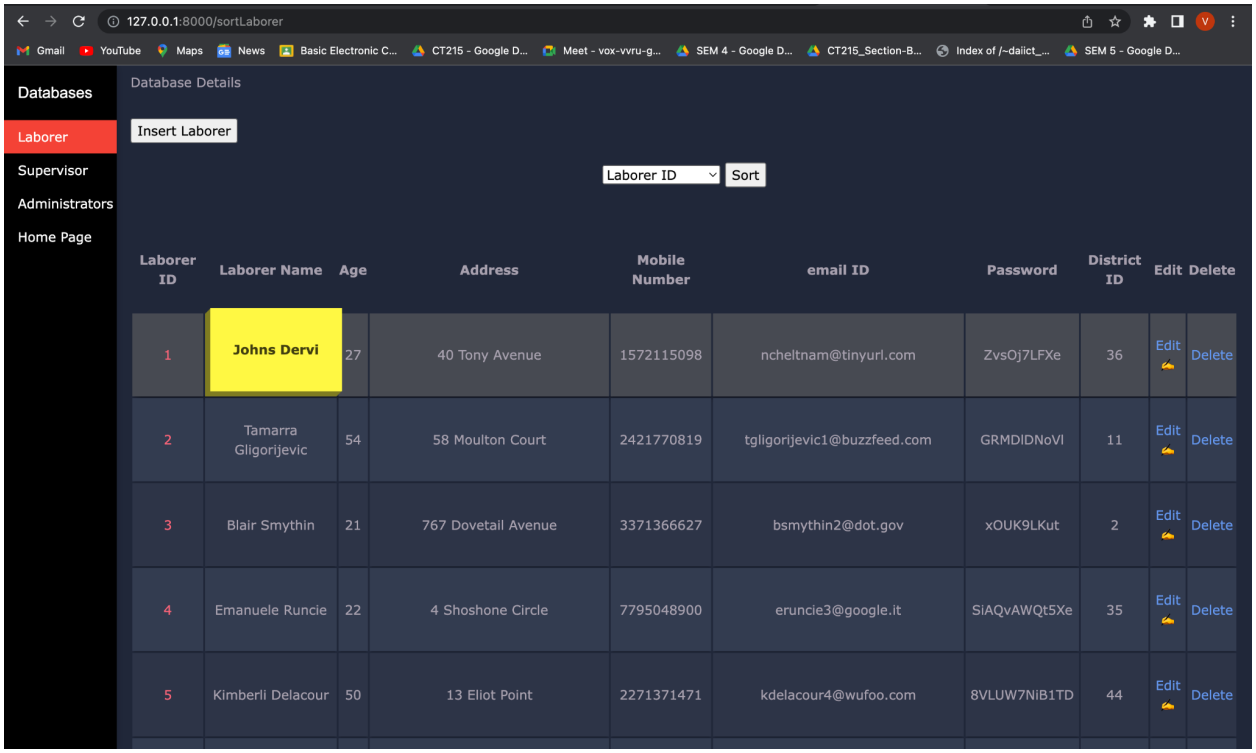
Before editing laborer ID 1



Editing name of laborer ID 1

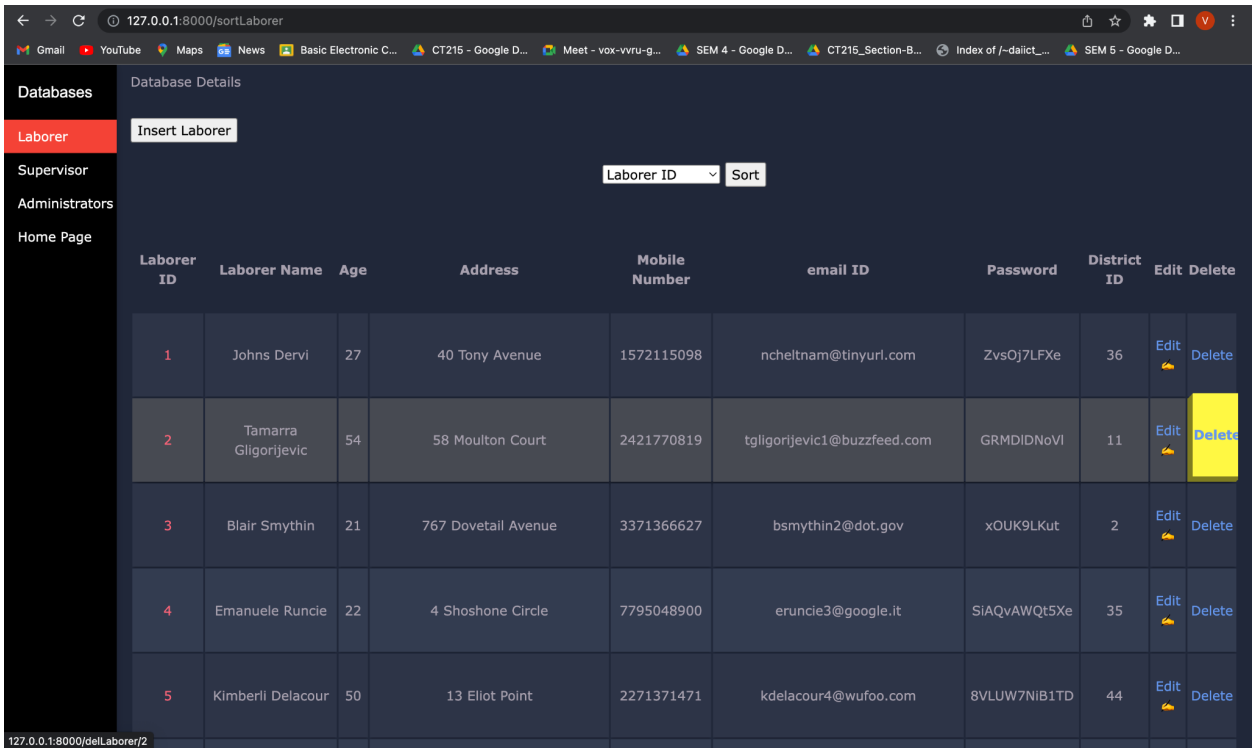


After editing the name of laborer ID 1

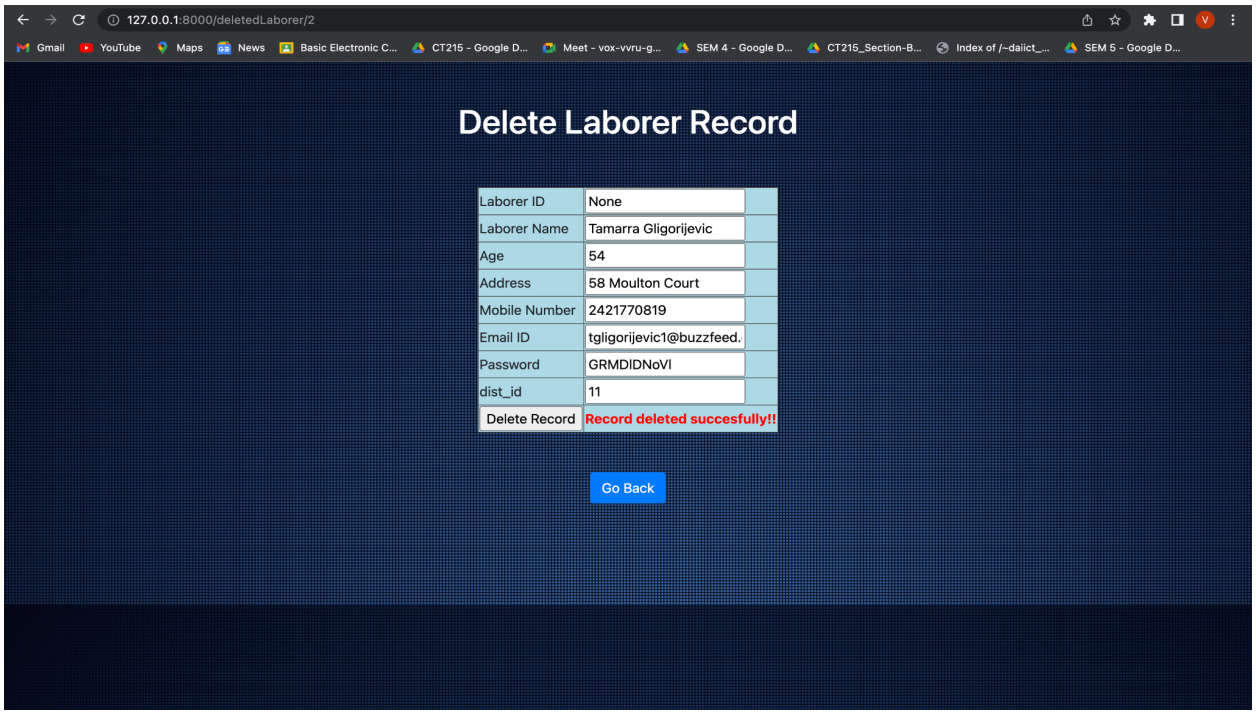


Delete Operation on tables

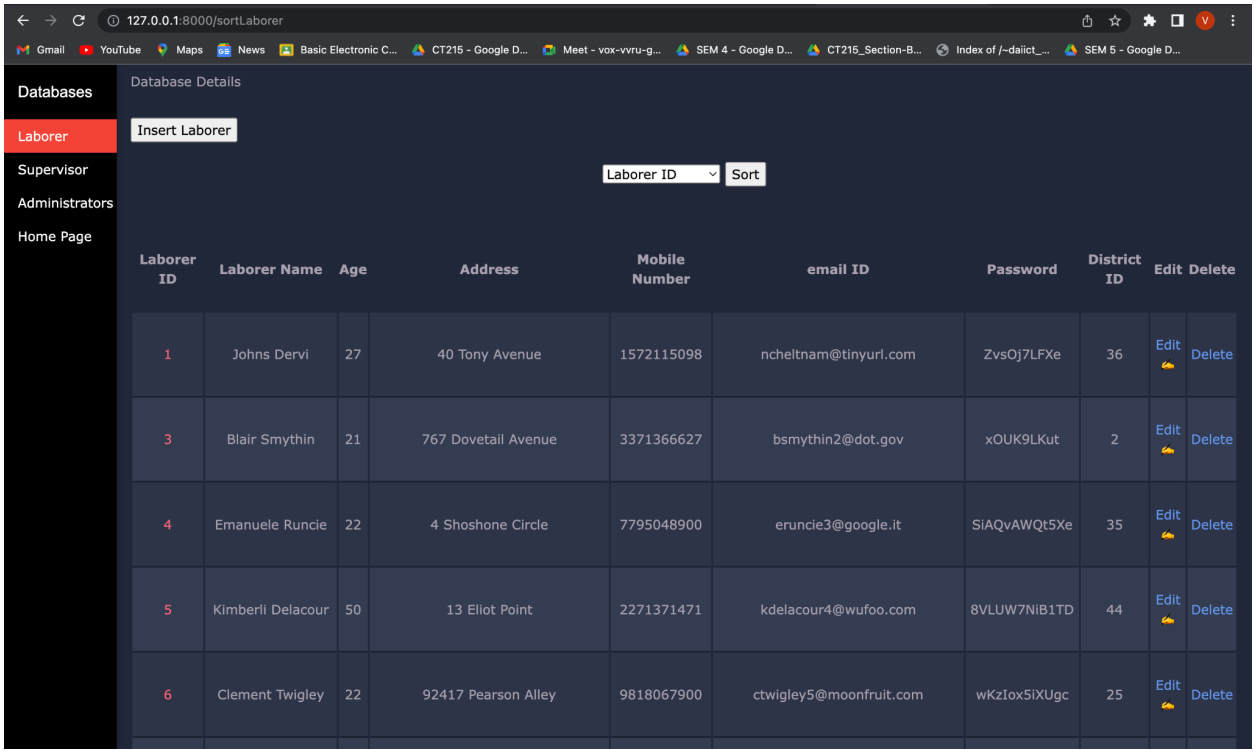
Before deleting laborer ID 2 records



Delete the record of laborer ID 2



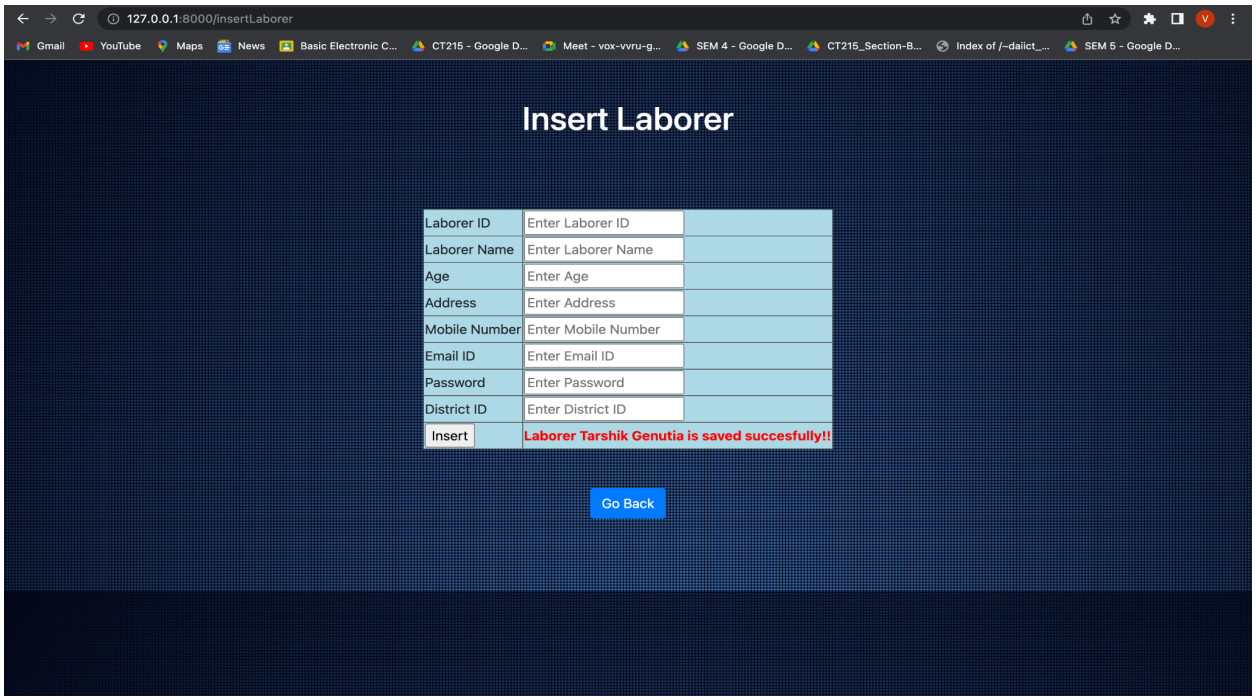
After Deleting data of laborer 2



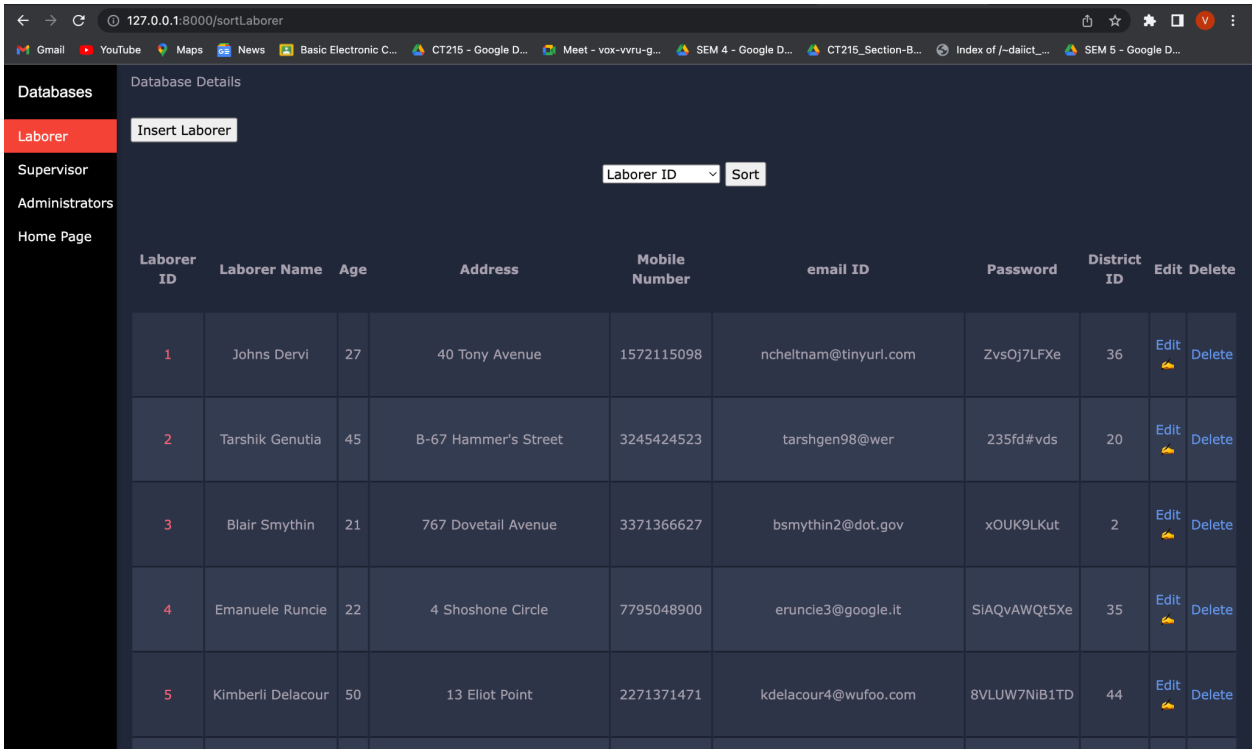
Laborer ID	Laborer Name	Age	Address	Mobile Number	email ID	Password	District ID	Edit	Delete
1	Johns Dervi	27	40 Tony Avenue	1572115098	ncheltnam@tinyurl.com	ZvsOj7LFXe	36	Edit	Delete
3	Blair Smythin	21	767 Dovetail Avenue	3371366627	bsmythin2@dot.gov	xOUK9LKut	2	Edit	Delete
4	Emanuele Runcie	22	4 Shoshone Circle	7795048900	eruncie3@google.it	SiAQvAWQt5Xe	35	Edit	Delete
5	Kimberli Delacour	50	13 Eliot Point	2271371471	kdelacour4@wufoo.com	8VLUW7NiB1TD	44	Edit	Delete
6	Clement Twigley	22	92417 Pearson Alley	9818067900	ctwigley5@moonfruit.com	wKzIox5iXUgc	25	Edit	Delete

Insert operation on tables

Inserting data for laborer ID 2 again



After inserting data for laborer ID 2

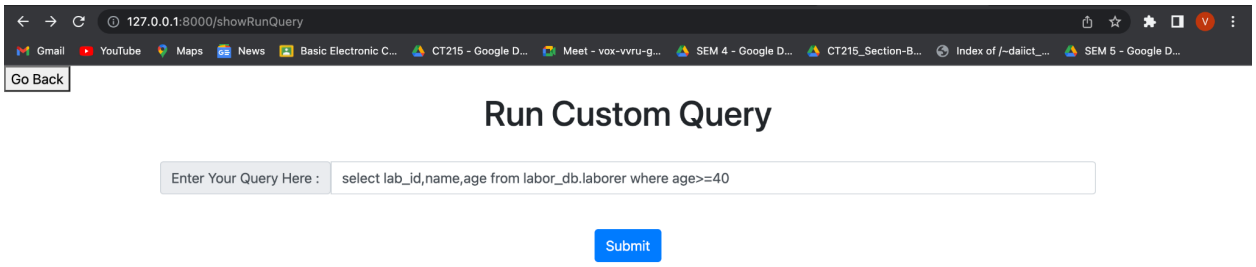


Laborer ID	Laborer Name	Age	Address	Mobile Number	email ID	Password	District ID	Edit	Delete
1	Johns Dervi	27	40 Tony Avenue	1572115098	ncheltnam@tinyurl.com	ZvsOj7LFXe	36	Edit	Delete
2	Tarshik Genutia	45	B-67 Hammer's Street	3245424523	tarshgen98@wer	235fd#vds	20	Edit	Delete
3	Blair Smythin	21	767 Dovetail Avenue	3371366627	bsmythin2@dot.gov	xOUK9LKut	2	Edit	Delete
4	Emanuele Runcie	22	4 Shoshone Circle	7795048900	eruncie3@google.it	SiAQvAWQt5Xe	35	Edit	Delete
5	Kimberli Delacour	50	13 Eliot Point	2271371471	kdelacour4@wufoo.com	8VLUW7NiB1TD	44	Edit	Delete

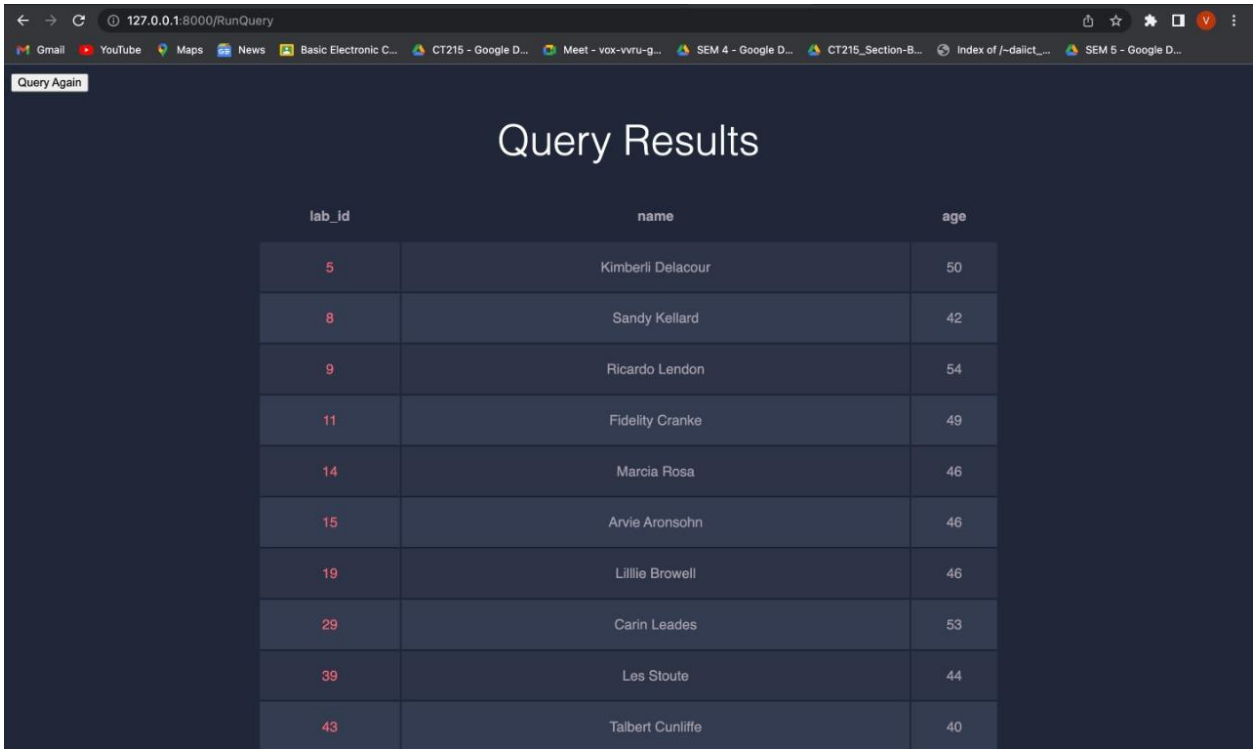
Running Custom Queries

⇒We can run any PostgreSQL query related to our database on our website itself. Here, we give some examples of our custom queries.

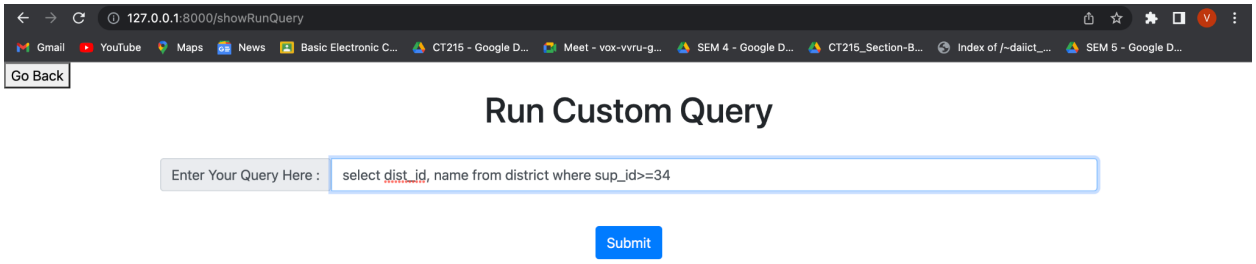
1. select lab_id,name,age from labor_db.laborer where age>=40



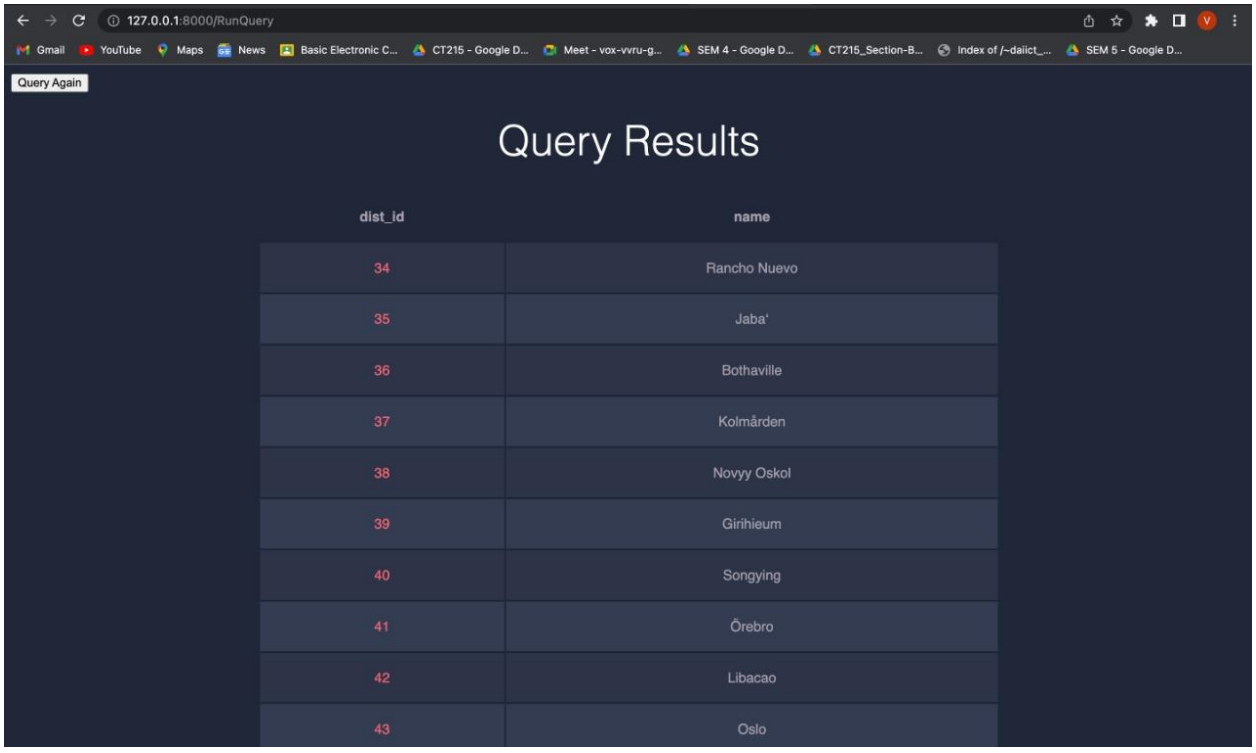
Result



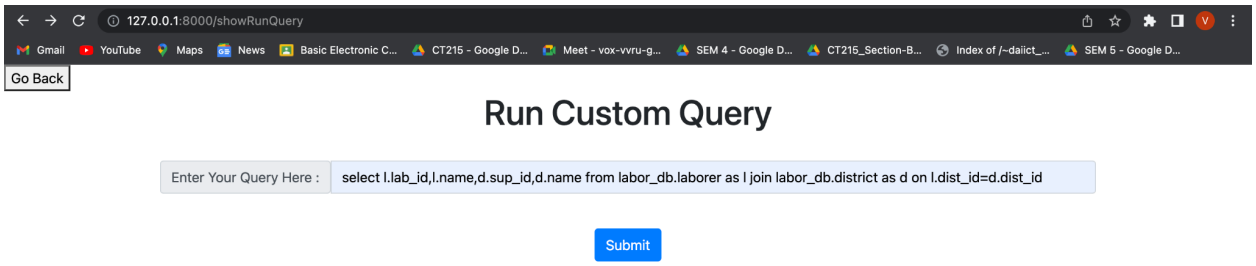
2. **Select dist_id, name from district where sup_id>=34**



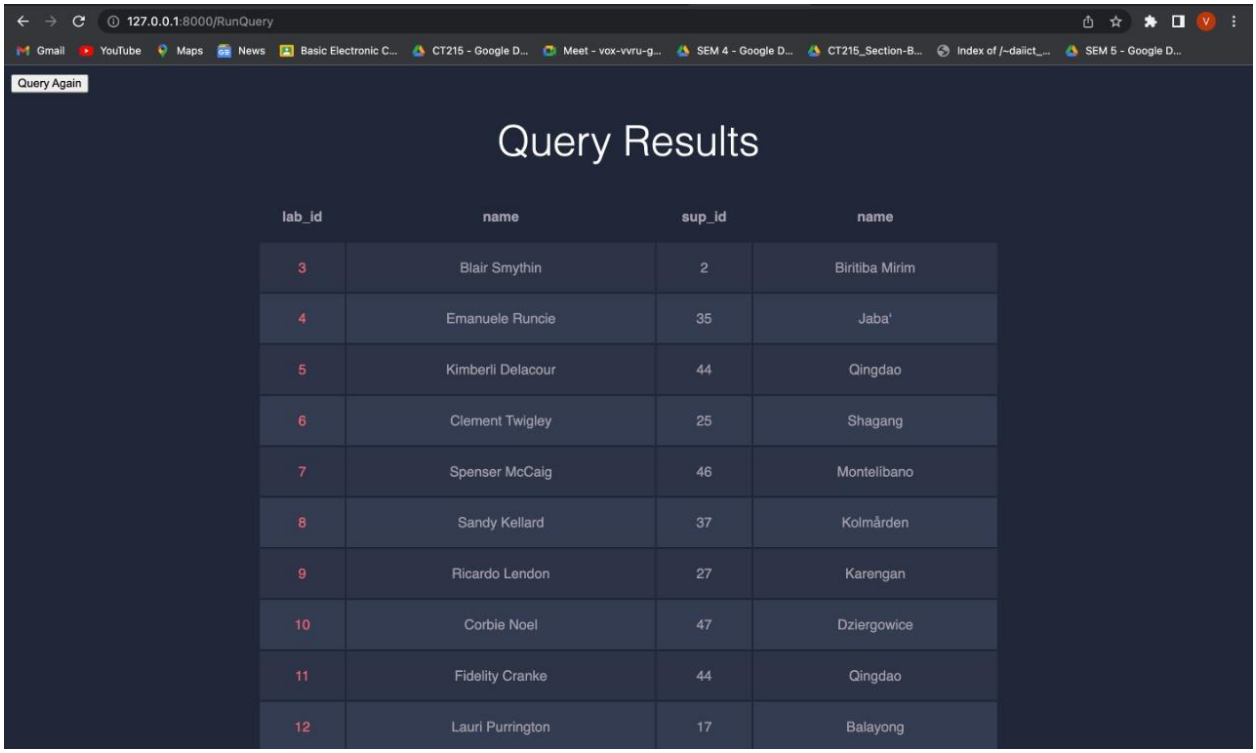
Result



3. **select l.lab_id,l.name,d.sup_id,d.name from labor_db.laborer as l join labor_db.district as d on l.dist_id=d.dist_id**



Result



lab_id	name	sup_id	name
3	Blair Smythin	2	Biritiba Mirim
4	Emanuele Runcie	35	Jaba'
5	Kimberli Delacour	44	Qingdao
6	Clement Twigley	25	Shagang
7	Spenser McCaig	46	Montelíbano
8	Sandy Kellard	37	Kolmården
9	Ricardo Lendon	27	Karengan
10	Corbie Noel	47	Dziergowice
11	Fidelity Cranke	44	Qingdao
12	Lauri Purrington	17	Balayong

If we submit an empty query then it will display an alert message ('Please fill in this field').